

**Universidad Carlos III de Madrid  
Escuela Politécnica Superior**



**INGENIERÍA DE TELECOMUNICACIÓN**

**Proyecto Fin de Carrera**

# **Sistema de Localización de Dispositivos Móviles Basada en Wireless LAN**

**Autora: Aurora Agudo de Carlos**

**Tutor: Julio Villena Román**

**Departamento de Ingeniería Telemática**

**Junio 2009**



**Título:** *Sistema de Localización de Dispositivos Móviles Basada en Wireless LAN*

**Autor:** Aurora Agudo de Carlos

**Tutor:** Julio Villena Román

## EL TRIBUNAL

Presidente:  
Iria Manuela Estévez Ayres

Secretario:  
Iván Vidal Fernández

Vocal:  
María Teresa Vicente Díez

Realizado el acto de defensa del Proyecto Fin de Carrera el día 16 de Junio de 2009 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de:

Fdo: Presidente

Fdo: Vocal

Fdo: Secretario



## Agradecimientos

Gracias a Julio, mi tutor, por darme la oportunidad de involucrarme en el mundo de las redes inalámbricas, ese desconocido por mi parte, y por la paciencia que ha tenido conmigo.

Gracias a mis padres y a mi hermano por su apoyo incondicional y por evitar que desistiera en el intento.

Gracias a David por estar siempre conmigo, dándome ánimos y prestándome su apoyo.

Gracias a Javi, ese gran amigo que me ha ayudado con sus innumerables consejos.



## Resumen

En los últimos tiempos se ha podido apreciar un gran auge en las tecnologías inalámbricas, así como una gran incursión en el mercado de los servicios de información y comunicaciones de los servicios de localización. Por ello surgen diversos proyectos de sistemas de localización en interiores, tanto en entornos de investigación, como sistemas comerciales. Estos sistemas son totalmente cerrados, no se puede modificar el algoritmo de localización que se haya elegido. Además en código abierto solo se pueden apreciar sistemas sobre Linux/Unix, cuando una gran parte de los dispositivos inalámbricos poseen S.O. de Microsoft.

Este Proyecto Fin de Carrera aborda la construcción de un Sistema de Posicionamiento totalmente escalable, que permita realizar localización de dispositivos en interiores con redes locales inalámbricas en el que se puedan evaluar distintos algoritmos de localización, al ser un sistema totalmente modular.

El resultado final de este Proyecto Fin de Carrera es un Sistema de Posicionamiento para cualquier entorno de interior (en casa, en la oficina, en universidades, en Centros Comerciales) que disponga de cobertura de red local inalámbrica (*WLAN*). El sistema permitirá tanto localizarse como ser localizado. Es decir, una persona que disponga de este sistema podrá situarse dentro de un edificio en el que no conozca su situación, y, a su vez, permitir ser localizado dentro del mismo. Dicho sistema podrá ser ejecutado en cualquier sistema operativo, ya sea Microsoft Windows o Linux/Unix, debido a su portabilidad. También podrá cambiarse el algoritmo de localización gracias a su modularidad.



## Abstract

Nowadays wireless technologies have experienced a great expansion, and they have conquered the market of information and communications services within the location services. Hence, a plethora of indoor location system projects arises, both in research environments and in commercial systems. These systems are completely closed. Their main drawback is the fact that the location algorithm can not be changed. In addition, open source systems are usually developed for Linux/Unix, although most of the wireless devices have Microsoft Operating System.

The Project addresses the construction of a positioning system fully scalable; the goal is to locate wireless devices in indoor environments. The user or operator is given the chance to select different location algorithms in a fully modular system.

The result of this Project is a positioning system for any indoor environment (at home, at the office, in the universities, at the commercial centers) that provide coverage of Wireless LAN (WLAN). The system allows the device to be located and to obtain its own position. For example, a person who has this system may be located within a building that does not know her/his situation, and, in turn, permits to be located within it. That system may be run on any operating system, either Microsoft Windows or Linux/Unix, because of its portability. The algorithm for locating devices may be easily changed due to the modularity of the system.



# Índice

<b>ÍNDICE DE FIGURAS .....</b>	<b>8</b>
<b>ÍNDICE DE TABLAS .....</b>	<b>10</b>
<b>1 INTRODUCCIÓN .....</b>	<b>11</b>
1.1 CONTEXTO.....	11
1.2 OBJETIVO .....	12
1.3 CONTENIDO .....	13
<b>2 ESTADO DEL ARTE .....</b>	<b>14</b>
2.1 POSICIONAMIENTO .....	14
2.1.1 CLASIFICACIÓN.....	15
2.1.1.1 POSICIONAMIENTO POR SATÉLITE .....	15
2.1.1.1.1 GPS .....	15
2.1.1.1.2 GALILEO .....	17
2.1.1.2 POSICIONAMIENTO BASADO EN REDES DE COMUNICACIONES.....	18
2.1.1.2.1 POSICIONAMIENTO EN REDES MÓVILES .....	18
2.1.1.2.2 POSICIONAMIENTO EN REDES LOCALES INALÁMBRICAS .....	19
2.1.1.2.3 POSICIONAMIENTO EN REDES AD-HOC.....	19
2.1.2 COMPARATIVA.....	20
2.2 REDES LOCALES INALÁMBRICAS .....	20
2.2.1 802.11.....	21
2.2.2 VENTAJAS E INCONVENIENTES.....	28
2.3 MODELOS DE SISTEMAS DE LOCALIZACIÓN.....	29
2.3.1 MODELOS EMPÍRICOS.....	29
2.3.2 MODELOS DE PROPAGACIÓN.....	30
2.4 PRINCIPALES MÉTODOS DE POSICIONAMIENTO EN SISTEMAS DE LOCALIZACIÓN WIFI .....	30
2.4.1 VECTOR DE POTENCIAS .....	30
2.4.2 TRIANGULACIÓN.....	31
2.4.3 TRILATERACIÓN.....	31
2.4.4 TÉCNICAS HEURÍSTICAS DE OPTIMIZACIÓN. ....	33
2.5 PROYECTOS DE SISTEMAS DE LOCALIZACIÓN Y SISTEMAS COMERCIALES. ....	34
2.5.1 RADAR.....	35
2.5.2 SKYHOOKWIRELESS .....	35
2.5.3 AMULET.....	36
2.5.4 HORUS .....	36
2.5.5 HALIBUT.....	36
2.5.6 LEASE .....	37
2.5.7 EKAHAU POSITIONING ENGINE .....	38
2.5.8 HERECast .....	39
2.5.9 PLACELAB.....	40
2.5.10 GOOGLE LATITUDE.....	41
<b>3 ESTUDIO DE VIABILIDAD DEL SISTEMA .....</b>	<b>45</b>
3.1 DESCRIPCIÓN DEL SISTEMA .....	45
3.1.1 ÁMBITO .....	45
3.1.2 DESCRIPCIÓN GENERAL.....	45
3.1.2.1 FUNCIONES DEL SISTEMA .....	45
3.1.2.2 CARACTERÍSTICAS DE LOS USUARIOS.....	47
3.1.2.3 RESTRICCIONES.....	47
3.1.3 REQUISITOS ESPECÍFICOS.....	47
3.1.3.1 REQUISITOS FUNCIONALES .....	47
3.1.3.2 REQUISITOS INTERFACES EXTERNOS .....	48
3.1.3.3 REQUISITOS RENDIMIENTO .....	48



3.1.3.4	REQUISITOS TECNOLÓGICOS .....	49
3.1.3.5	SEGURIDAD .....	49
3.1.4	ANÁLISIS DE ALTERNATIVAS .....	49
3.1.4.1	DESCRIPCIÓN GENERAL .....	49
3.1.4.2	ALTERNATIVAS .....	50
<b>4</b>	<b>DISEÑO.....</b>	<b>51</b>
4.1	OBJETIVO DEL SISTEMA .....	51
4.1.1	CONTEXTO .....	51
4.1.2	CASOS DE USO .....	53
4.2	DISEÑO LÓGICO .....	55
4.2.1	ESTRUCTURA ESTÁTICA DEL SISTEMA .....	55
4.2.1.1	SISTEMA .....	55
4.2.1.2	COMPONENTES .....	56
4.2.1.3	INTERFACES .....	57
4.2.2	ESTRUCTURA ESTÁTICA DEL CLIENTE .....	58
4.2.2.1	CLIENTE .....	58
4.2.2.2	COMPONENTES .....	58
4.2.2.3	INTERFACES .....	59
4.2.3	COMPORTAMIENTO DINÁMICO .....	60
4.2.3.1	ESCENARIOS – PROTOCOLO DE COMUNICACIÓN .....	60
4.2.4	BASE DE DATOS .....	80
4.3	RESTRICCIONES .....	81
<b>5</b>	<b>IMPLEMENTACIÓN.....</b>	<b>82</b>
5.1	DECISIONES PREVIAS .....	82
5.1.1	EL LENGUAJE DE PROGRAMACIÓN .....	82
5.1.2	ALGORITMO DE LOCALIZACIÓN .....	83
5.1.3	BASE DE DATOS .....	83
5.2	DIAGRAMA DE CLASES UML .....	84
5.2.1	DIAGRAMA DE CLASES UML DEL SERVIDOR .....	84
5.2.2	DIAGRAMA DE CLASES UML DEL CLIENTE .....	88
5.3	BASE DE DATOS .....	90
5.4	RESTRICCIONES .....	91
<b>6</b>	<b>PRUEBAS Y RESULTADOS.....</b>	<b>92</b>
6.1	ENTORNO DE PRUEBAS .....	92
6.2	CLIENTE DE PRUEBAS .....	93
6.2.1	CLIENTE TEXTO .....	93
6.2.2	CLIENTE GRÁFICO .....	94
6.2.3	GAUGE .....	96
6.2.4	USER/MOBILE .....	100
6.2.5	MONITOR .....	101
6.3	RESULTADOS DE CALIBRACIÓN .....	102
<b>7</b>	<b>CONCLUSIONES Y LÍNEAS FUTURAS.....</b>	<b>103</b>
7.1	CONCLUSIONES .....	103
7.2	LÍNEAS FUTURAS .....	104
<b>8</b>	<b>BIBLIOGRAFÍA Y REFERENCIAS .....</b>	<b>106</b>
<b>APÉNDICE A: GLOSARIO .....</b>		<b>109</b>
<b>APÉNDICE B: MANUAL DE DESPLIEGUE DEL SERVIDOR .....</b>		<b>116</b>
<b>APÉNDICE C: MANUAL DE USUARIO DEL CLIENTE GRÁFICO DE PRUEBA.....</b>		<b>117</b>
<b>APÉNDICE D: MANUAL DE USUARIO DEL CLIENTE TEXTO DE PRUEBA. ....</b>		<b>119</b>
<b>APÉNDICE E: MANUAL DE CREACIÓN DE CLIENTE PARA EL SISTEMA.....</b>		<b>120</b>

## Índice de Figuras

Fig 1: Descripción del sistema de satélites GPS.....	16
Fig 2: Ejemplo de Red Ad-Hoc. ....	24
Fig 3: Ejemplo de Red en Modo Infraestructura. ....	25
Fig 4: Ejemplo de sistema con Controlador de Acceso.....	26
Fig 5: Ejemplo de Cálculo de Método Vector Potencia. ....	31
Fig. 6: Trilateración–Detección de 1 AP. ....	32
Fig 7: Trilateración- Detección de 2 APs. ....	32
Fig 8: Trilateración- Detección de 3 APs. ....	33
Fig. 9: Diagrama de contexto.....	51
Fig 10: Casos de Uso. ....	53
Fig 11: Arquitectura del Sistema. ....	56
Fig 12.- Arquitectura del Cliente. ....	58
Fig 13: Establecimiento de Conexión realizado con éxito. ....	61
Fig 14: Establecimiento de conexión con errores.....	61
Fig 15: Establecimiento de conexión con errores.....	62
Fig 16: Inital (Configuración) realizada con éxito.....	63
Fig 17: Inital (Configuración) realizada con errores.....	63
Fig 18: Inital (Configuración) realizada con errores.....	64
Fig 19: Cal (Calibración) realizada con éxito.....	65
Fig 20: Cal (Calibración) realizada con errores.....	65
Fig 21: Cal (Calibración) realizada con errores.....	66
Fig 22: Cal (Calibración) realizada con errores.....	66
Fig 23: delCal (Borrado de Calibración) realizada con éxito.....	67
Fig 24: delCal (Borrado de Calibración) realizada con errores.....	67
Fig 25: Zone (Obtención de número de zonas) realizada con éxito.....	68
Fig 26 Zone (Obtención de número de zonas) realizada con errores.....	68
Fig 27: List realizada con éxito.....	69
Fig 28: List con errores.....	69
Fig 29: LOC realizada con éxito.....	71
Fig 30: Loc realizada con errores.....	71
Fig 31: Loc con errores.....	72
Fig 32: Loc con errores.....	72
Fig 33: Loc con errores.....	73
Fig 34: SETLOC realizado con éxito.....	74
Fig 35: SetLoc realizado con errores.....	74
Fig 36: SetLoc con errores.....	75
Fig 37: GetAllLoc realizado con éxito.....	76
Fig 38: GetAllLoc con errores.....	77
Fig 39: GetAllLoc con errores.....	77
Fig 40: GetAllLoc con errores.....	77
Fig 41: GetAllLoc con errores.....	78
Fig 42: GetAllLoc con errores.....	79
Fig 43: Modelo Conceptual de la Base de Datos.....	80
Fig 44: Diagrama de Clases del Servidor.....	84
Fig 45: Conexión a Base de Datos.....	85





Fig 46: Clase WiLBSMultiServerT .....	86
Fig 47: Clase PowVector. ....	86
Fig 48: WiLBSDData clase. ....	87
Fig 49: Diagrama de clases del Cliente Gráfico. ....	88
Fig 50: Diagrama de clases del Cliente User. ....	89
Fig 51: Diagrama de clases del Cliente Texto. ....	89
Fig 52: Base de datos .....	90
Fig 53: Ejemplo de Mapa de Localización. ....	93
Fig 54: Pantalla Inicial. ....	94
Fig 55: Pantalla inicial con datos correctos. ....	95
Fig 56: Pantalla inicial – Error en el usuario o en el password. ....	95
Fig 57: Pantalla de listado de localizaciones. ....	96
Fig 58: Pantalla de configuración de las localizaciones .....	97
Fig 59: Pantalla de calibración de la localización.....	98
Fig 60: Pantalla de borrado de la localización. ....	99
Fig 61: Pantalla de Posicionamiento del usuario. ....	100
Fig 62: Icono de la aplicación.....	101
Fig 63: User- desactivar opción de “localizable” .....	101
Fig 64: Monitor – Localización de usuarios .....	101



## *Índice de Tablas*

Tabla 1: Componente “SERVER” .....	56
Tabla 2: Componente “DB ACCESS” .....	56
Tabla 3: Componente “CLASSIFIER” .....	57
Tabla 4: Interfaz “SOCKET” .....	57
Tabla 5: Interfaz SQL. ....	57
Tabla 6: Componente “PRESENTATION” .....	58
Tabla 7: Componente “WIFI ACCESS” .....	59
Tabla 8: Interfaz SOCKET. ....	59
Tabla 9: Interfaz con ‘Wifipos’ .....	59

# 1 Introducción

## 1.1 Contexto

En los últimos años las redes inalámbricas han sufrido una fuerte expansión tanto en interiores como en zonas urbanas. Esto ha dado lugar a la aparición de una serie de servicios asociados a dichas redes debido a las funcionalidades que aprovechan el gran número de redes desplegadas y a su sencillez de uso. Al mismo tiempo se puede ver cómo el uso del *GPS* ha ido creciendo, siendo ya casi un elemento imprescindible para el transporte de viajeros y mercancías por carretera.

En un mundo en que las posibilidades de conexión aumentan año tras año, en el que la mayor parte de la población dispone de acceso a redes inalámbricas, existe un gran potencial social y económico en los servicios basados en la localización, que, por distintas circunstancias, aún no han sido muy explotados.

De entre los servicios basados en localización, los más desarrollados, son, con mucha diferencia los de las redes móviles (*GSM*, *CDMA*, *UMTS*). Estos servicios, sin embargo, no hacen uso de las redes inalámbricas de área local y de área urbana, cuya presencia ha crecido exponencialmente en los últimos años, y que presentan grandes ventajas: facilidad de acceso con múltiples dispositivos, eficiencia económica, control de acceso, etc.

A raíz de la incursión en el mercado de las redes inalámbricas y del *GPS* como ya se ha mencionado, nace la necesidad de desarrollar servicios de localización en interiores, ya que la señal de *GPS* no llega dentro de los edificios (no es útil para entornos denominados ‘indoors’, de interior). Debido a esto han surgido diversos proyectos tanto en el entorno de investigación como comerciales. Habiendo sido diseñados para que su funcionamiento solo pueda llevarse a cabo en dispositivos con un sistema operativo en concreto, Linux o Windows; así como para que realicen localización basándose en un algoritmo de localización determinado, sin posibilidad de cambio en el futuro. Siendo más frecuente ver que los proyectos comerciales se enfocan más a entornos con software propietario como es Microsoft Windows y los de investigación más a entornos de software abierto como es Linux.

De entre todas las redes inalámbricas de área local o urbana, un buen candidato de trabajo son las redes WLAN, más comúnmente denominadas WiFi, por sus claras ventajas: facilidad de despliegue, inmensa cantidad de terminales disponibles (la gran mayoría de ordenadores actuales, móviles y videoconsolas soportan *WiFi*, por citar solo algunos ejemplos), escaso coste (consecuencia de la fabricación masiva de chips que ofrecen funcionalidad *WiFi*) y amplio número de redes ya existentes, que nos permiten hablar de universalidad de *WiFi* como protocolo inalámbrico para redes locales.



## **1.2 Objetivo**

Este proyecto pretende centrarse en los servicios de localización en las redes inalámbricas. Así, partiendo de un análisis de las distintas soluciones disponibles hoy en día para los servicios basados en localización, se desarrollará una plataforma a través de la cual distintos dispositivos móviles podrán conocer su localización o hacer posible su localización por otros usuarios. Se creará, así, por tanto, la infraestructura necesaria para el desarrollo de aplicaciones que ofrezcan los servicios basados en localización – a cada una de estas aplicaciones le bastaría con comportarse como clientes en el que esquema desarrollado en este proyecto para poder ofrecer servicios a sus usuarios. Esta infraestructura ha de ser lo más genérica, universal, eficiente y fácil de desplegar que sea posible. Además, se pretende que el sistema sea ampliable en el futuro, pudiendo evolucionar al mismo ritmo que las redes inalámbricas, por lo que se busca el diseño de una plataforma totalmente modular, escalable y multiplataforma.

A la vista de la utilidad de los servicios de localización y de las características de los proyectos existentes, se ha tomado como objetivo el realizar un sistema multiplataforma y totalmente modular para que sea posible cambiar el algoritmo de posicionamiento; y así poder evaluar distintos algoritmos de localización sobre un mismo sistema. Por lo tanto, el sistema a realizar podrá correr en cualquier plataforma y a él se podrán conectar usuarios que posean cualquier tipo de sistema operativo. Así se desarrolla una plataforma de gran usabilidad y totalmente escalable, en la que se puedan alternar y estudiar algoritmos de localización.

Debido a la expansión de las redes de área local inalámbricas en interiores, es bastante habitual encontrar lugares con gran cobertura de las mismas, debido a la gran cantidad de puntos de acceso. Por esta razón se ha identificado la utilidad de un sistema de localización basado en redes WiFi para interiores similar al *GPS* que funciona en exteriores. Este sistema debe ser fácil de desplegar en casi cualquier lugar: en hogares, oficinas, espacios públicos como universidades, colegios, oficina del INEM, centros de ocio como Parques Temáticos, Grandes Centros Comerciales, así como para Centros de Congresos o Visitas Culturales.

## **1.3 Contenido**

Este Proyecto comienza dando una visión global de los métodos existentes para el posicionamiento así como el estado del arte de dichos sistemas. Es extremadamente importante conocer el ecosistema en que se mueve el desarrollo del proyecto: qué tipo de servicios de localización existen, sobre qué redes se prestan, etc. Estos sistemas se desarrollan con más detalle en el **Capítulo 2 – Estado del arte**.

Después de haber descrito el marco en el que se desarrolla este proyecto, se realiza un Estudio de Viabilidad del mismo que se encuentra en el **Capítulo 3 – Estudio de Viabilidad del Sistema**. En él se analizan las funcionalidades deseadas, los requisitos, y las restricciones. Se termina con un análisis de las alternativas de diseño de la plataforma..

Una vez definidas las características y requisitos del sistema, en el **Capítulo 4 – Diseño** se realiza el diseño para la plataforma seleccionada en el estudio del Capítulo anterior

El siguiente **Capítulo 5 – Implementación** trata sobre la implementación y los detalles de bajo nivel del sistema,. La realización de las pruebas (“testing”) pertinentes a dicho sistema está descrita en el **Capítulo 6 – Pruebas y Resultados**. Dichas pruebas se han llevado a cabo usando uno de los algoritmos de posicionamiento más utilizados y que mejores resultados está obteniendo hasta el momento.

Para finalizar, en el **Capítulo 7 – Conclusiones y Líneas Futuras**, donde se exponen las conclusiones que se obtienen del diseño y de las pruebas, y se recoge un breve análisis de las líneas futuras que se pueden considerar para dichos sistemas.

## ***2 Estado del Arte***

Para poder llevar a cabo la implementación de este proyecto, será necesario introducirse primero en el mundo del Posicionamiento y de las redes inalámbricas. Para ello, en este apartado se intentará dar una visión global del “background” tecnológico que va a influir en la realización de la Plataforma para estudio de algoritmos de localización en redes Wireless (*WiFi*).

En primer lugar, se verá el significado de ‘posicionamiento’, así como los distintos entornos en los que se pueden encontrar sistemas de este tipo, dando una pequeña clasificación e introducción al respecto.

Después de haber descrito con detalle el mundo del posicionamiento, se particularizará para el entorno concreto en el que se centra el Proyecto: el funcionamiento de las Redes Locales Inalámbricas y los sistemas de localización que se basan en ellas.

Después se podrá apreciar un breve resumen de los algoritmos de posicionamiento más extendidos, su situación actual y la tasa de error que suelen tener.

Se concluye este apartado analizando algunos de los sistemas de localización *WiFi* existentes en el mercado. Como se puede observar posteriormente, la gran mayoría de ellos han nacido en el seno de la investigación de universidades, llegando en algunos casos a ser comercializados.

### ***2.1 Posicionamiento***

Los sistemas de posicionamiento [21], o también conocidos como sistemas de localización, se extienden cada vez más y empiezan a ser imprescindibles en la vida cotidiana al igual que ocurre con el móvil, que ya no se puede salir de casa sin él.

El rápido avance de la tecnología, así como de los sistemas inteligentes ha hecho posible el desarrollo sencillo de sistemas con funciones de localización, es decir, con la capacidad de determinar la ubicación del dispositivo en un área determinada, para diversas funcionalidades. Tanto si se trata de sistemas de localización en los que el usuario o terminal es el encargado de obtener su propia posición (como por ejemplo el sistema navegador de los vehículos o la orientación de usuarios móviles), como de sistemas de gestión centralizados (seguimiento de usuarios en las redes, móviles o *WLAN*).

Estos sistemas permiten la instalación de multitud de servicios al usuario, como información de la parada de autobús más cercana, ayuda a los consumidores para encontrar establecimientos, localización de menores...

### **2.1.1 Clasificación**

La implantación de sistemas de posicionamiento comenzó con sistemas globales, basados en la señal de satélites repartidos en el mundo, y ha continuado hasta adaptarse a todo tipo de ambientes y tareas, tras el crecimiento en los diversos tipos de área local existentes. Por lo que se puede clasificar el desarrollo de sistemas de posicionamiento en dos grandes clases:

- Posicionamiento por Satélite
- Posicionamiento basado en redes de comunicaciones (o en redes de área local y metropolitana).

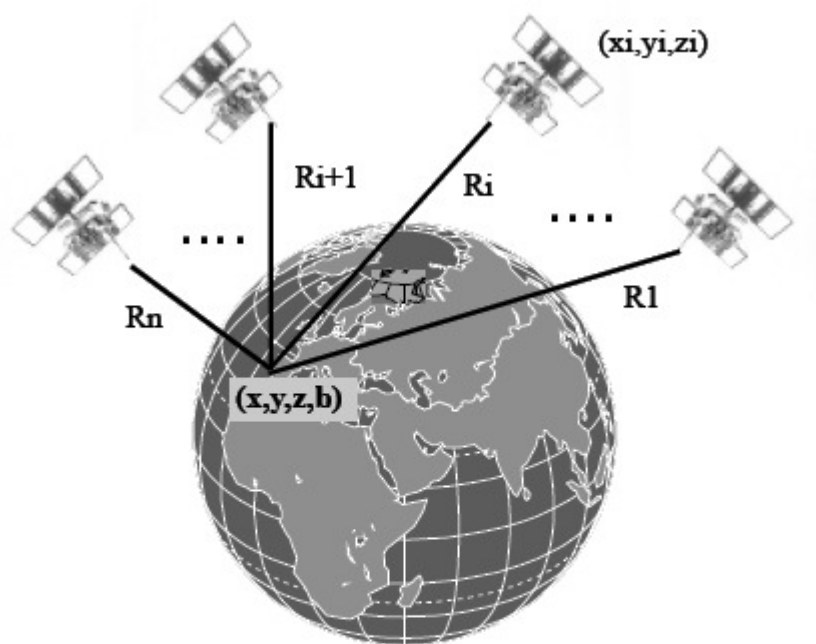
#### **2.1.1.1 Posicionamiento por Satélite**

El sistema de posicionamiento pionero es el conocido como Sistema de Posicionamiento Global (GPS – Global Positioning System) [31]. Fue desarrollado en el ámbito militar (por la empresa NAVSTAR), como herramienta para la estimación de Posición, Velocidad y Tiempo de vehículos. Entró en servicio en el año 1978. Poco tiempo después surge el sistema ruso GLONASS, de similares características, también para uso militar. A pesar de su origen militar, paulatinamente han ido apareciendo aplicaciones civiles, que han tenido un crecimiento espectacular en los últimos años.

La Unión Europea decidió que era de importancia estratégica el contar con un sistema de navegación por satélite propio, dependiente de una autoridad civil. De esta forma no tendría que depender de un sistema militar de los Estados Unidos. Este fue el origen del sistema Galileo. Este sistema actualmente se encuentra en desarrollo y cuenta con dos satélites para realizar pruebas.

##### **2.1.1.1.1 GPS**

*GPS* es un sistema de navegación por satélite [26] . en el que utiliza conjuntamente una red de ordenadores y una constelación de 24 satélites para determinar posición en tres dimensiones (la altitud, longitud y latitud) de cualquier objeto cerca de la superficie terrestre y para sincronizar los receptores con un tiempo común.



$\{R_i\}$ : Pseudodistancia

$(x_i, y_i, z_i)$ : posición conocida de los satélites

$$R_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} - b$$

Si  $N \geq 4$  se obtiene  $(x, y, z)$  y la desviación del reloj  $b$

Fig 1: Descripción del sistema de satélites GPS.

El principio de funcionamiento se muestra en la [Fig 1]. La posición de un punto cualquiera en un espacio de coordenadas  $(x, y, z)$  se obtiene partiendo del cálculo de las distancias del punto a un mínimo de tres satélites cuya localización es conocida. La distancia entre el usuario receptor *GPS* y un satélite se mide multiplicando el tiempo de vuelo de la señal emitida desde el satélite por su velocidad de propagación. Para medir el tiempo de vuelo de la señal de radio es necesario que los relojes de los satélites y de los receptores estén sincronizados, pues deben generar simultáneamente el mismo código. Ahora bien, mientras los relojes de los satélites son muy precisos los de los receptores son osciladores de cuarzo de bajo coste y por tanto imprecisos. Las distancias con errores debidos al sincronismo se denominan *pseudodistancias*. La desviación en los relojes de los receptores añade una incógnita más que hace necesario un mínimo de cuatro satélites para estimar correctamente las posiciones.

Este sistema de navegación por satélite tiene el inconveniente de que no funciona en interiores porque la señal es muy débil o nula. La aplicación civil que se ha usado con mayor éxito ha sido la navegación de vehículos por carreteras.

Debido a la inaccesibilidad en los entornos urbanos e interiores, se ha desarrollado una serie de extensiones de sistema *GPS*, para obtener un mejor rendimiento. Estas extensiones se apoyan en otras infraestructuras de red que complementan a los satélites:



- Utilización de las redes móviles para enviar información adicional sobre las órbitas de los satélites y correcciones de las señales a través de un receptor de datos. Esta técnica se denomina *GPS* Asistido ( *A-GPS*, **Assisted Global Positioning System**).
- El *D-GPS* (Differential *GPS*), o *GPS* diferencial, es un sistema que proporciona a los receptores de *GPS* correcciones de los datos recibidos de los satélites *GPS*, con el fin de proporcionar una mayor precisión en la posición calculada.
- Otras técnicas mejoradas para aumentar la precisión a pocos metros, y algunos incluso a unos pocos milímetros.

En el campo de la ingeniería civil, se ha convertido en una herramienta indispensable para profesionales y técnicos para la determinación de posiciones y realización de levantamientos topográficos con rapidez y precisión.

### 2.1.1.1.2 GALILEO

Galileo [9] es el equivalente europeo al *GPS*, que se está desarrollando en estos momentos y del cual hay dos satélites en prueba orbitando, *Giove-A* y *Giove-B*. Es un Sistema Global de Navegación por Satélite (*GNSS*) para evitar la dependencia de *GPS* y *GLONASS*, a la vez que será un sistema interoperable con estos.

Proporcionará servicios de radionavegación así como de posicionamiento en el espacio, dividido en 5 servicios:

- Servicio abierto (Open Service – *OS*), servicio público gratis.
- Servicio para aplicaciones críticas ( *Safety of Live -SoL*)  
El *SoL* mediante pago está pensado para sistemas de aviación, proporcionando más integridad (es un sistema que avisa de si las medidas son o no fiables).
- Servicio público regulado (Public Regulated Service – *PRS*) para policía (con mayor integridad y precisión).
- Servicio de búsqueda y salvamento (Search and Rescue Service – *SAR*).
- Servicio Comercial (Commercial Service-*CS*)

El receptor usará satélites de distintas constelaciones para el cálculo de su ubicación en el espacio. Proporcionará ubicación en tiempo real con una precisión de metros gracias a las dos frecuencias en las que trabaja en su versión estándar.

El sistema Galileo estará formado por una constelación mundial de 30 satélites en órbita terrestre media distribuidos en tres planos inclinados con un ángulo de  $56^\circ$  hacia el ecuador, a 23.616 Km. de altitud. Estos satélites se encuentran ligeramente más inclinados hacia los polos que los de la malla *GPS* para proporcionar mejor cobertura en áreas cercanas a los polos pues allí es donde menos cobertura proporciona *GPS*.

### 2.1.1.2 *Posicionamiento basado en redes de comunicaciones*

Los sistemas de posicionamiento basado en redes [4], a su vez pueden clasificarse en tres tipos:

1. Posicionamiento en redes móviles (*CDMA*, *GSM* y otras bandas).
2. Posicionamiento en redes inalámbricas locales (*WLAN*).
3. Posicionamiento por sensores en redes ad-hoc.

#### 2.1.1.2.1 *Posicionamiento en redes móviles*

Basándose en la propia estructura de éstas redes, se puede ver cómo para su propio funcionamiento es necesario saber la celda en la que se encuentra el usuario, por lo que puede dar una primera aproximación a su situación.

En estos sistemas se usan sus principales características de arquitectura combinados o no con algoritmos de posicionamiento para encontrar la ubicación del usuario [29]:

- Identificación de celdas (*CELL ID*)

Se identifica la celda desde la que está emitiendo el móvil donde tiene cobertura. Sirve para ubicar todo tipo de dispositivos móviles en redes *GSM*, *GPRS*, *UMTS* y *CDMA*.

- Dirección o ángulo de llegada (*AOA*, Angle Of Arrival)

Este método utiliza antenas multiarray situadas en la estación base para determinar el ángulo de la señal incidente. Si un terminal que transmite una señal está en la línea de vista directa (*LOS*, Line Of Sight), la antena multiarray puede determinar de qué dirección viene la señal.

- En función de la diferencia de tiempos de llegada perfeccionado (*E-OTD*), usado en *GSM* mediante el algoritmo de triangulación.  
Se usa en *GSM* y *GPRS*, se necesitan unos receptores especiales y la instalación de un software específico en los terminales móviles. Los receptores y los móviles realizan medidas de las señales procedentes de tres o más estaciones base periódicamente. Y usan las diferencias de tiempo en la llegada de la señal entre 2 puntos, que combinadas producen unas líneas hiperbólicas que se intersectan en el lugar donde se encuentra el terminal.
- Diferencia del tiempo de llegada en el enlace ascendente. (*U-TDOA*).

Usa la diferencia entre los tiempos de llegada de la señal procedente del terminal móvil a distintos pares de estaciones base para calcular la posición.

- En función del tiempo absoluto de llegada (*TDOA*, Time Difference of Arrival).
- Huella multitrayecto (*MF*, Multipath Fingerprint,).

Calcula la diferencia de caminos debido al retardo de las señales.

- Trilateración avanzada de enlace hacia delante (*AFLT*, Advanced Forward Link Trilateration).

Solo se usa en redes *CDMA*, aprovechando el sincronismo con el que operan y las técnicas de *TDOA*.

- Sistema de posicionamiento Global avanzado (*AGPS*). Haciendo uso de los sistemas *GPS* anteriormente mencionados a través de *UMTS*.

Usan receptores de referencia que recogen información de navegación y datos de corrección diferencial para los satélites *GPS* que están en la zona de cobertura del servidor de localización.

### **2.1.1.2.2 Posicionamiento en redes locales inalámbricas**

El posicionamiento en redes inalámbricas locales (*WLAN*) [32] se basa en la infraestructura *LAN* inalámbrica y usa las mediciones de la potencia de la señal (abreviado como *SS*, Signal Strength) de los puntos de acceso (*AP*) de *LAN* inalámbrica o puentes para calcular la ubicación física del dispositivo de destino (*PDA*, ordenador portátil...) con la tarjeta *WLAN* – adaptador de red. La técnica también se conoce como ‘*location fingerprinting*’. Es útil en la mayoría de los ambientes interiores, donde las técnicas de *GPS* no cumplen los requisitos de las aplicaciones, pues la señal que llega es muy débil. Sin embargo, también puede trabajar para ambientes al aire libre, donde es probable que pueda superar la precisión de los dispositivos *GPS*, debido a que la señal percibida es mucho más potente.

Este tipo de posicionamiento es en el que se engloba el proyecto que se está llevando a cabo, por lo que en secciones posteriores se ampliará la información de este apartado.

### **2.1.1.2.3 Posicionamiento en redes Ad-Hoc**

Por último se encuentran las redes Ad-Hoc de posicionamiento basadas en sensores. Usan la propagación de señales a corta distancia para determinar la ubicación del usuario móvil con una precisión que puede llegara hasta la cobertura que da la señal de corto alcance en emisores de infrarrojos, ultrasonidos...

Ejemplos de este tipo de redes son:

- Bluetooth
- IrDA
- Active Badge
- Cricket

### **2.1.2 Comparativa**

Si se comparan los sistemas de redes locales de posicionamiento con los de posicionamiento global, se puede observar cómo las redes locales inalámbricas aventajan considerablemente al *GPS* en localización en entornos urbanos e interiores. Pues en entornos urbanos la señal de satélite se pierde bastante debido a grandes edificios, barreras arquitectónicas, por lo que el *GPS* fracasa en dichos entornos.

*GPS* es capaz de determinar los grados de longitud y latitud del usuario con una precisión de 8 a 10 metros, mientras que los sistemas en *WLAN* también obtienen la altura, con el precio de cometer un error de precisión superior, en torno a 10-20 metros.

En entornos interiores el canal inalámbrico es muy ruidoso debido a los elementos que se encuentran en los edificios y la señal se ve afectada por el efecto multitrayecto (reflexión, refracción y difracción). *GPS* no tiene este problema y por este motivo un sistema de localización basado en *WiFi* tiene menos precisión. Para resolver el problema del multitrayecto se aplican técnicas de entrenamiento en los dispositivos antes de estimar posiciones, cosa que no es necesaria en *GPS*.

Un sistema inalámbrico también reduce los costes de implementación de los servicios basados en localización, ya que no requiere una infraestructura tan cara como la de *GPS* o la de las redes móviles, y no se necesitan receptores específicos.

## **2.2 Redes Locales inalámbricas**

Las redes locales inalámbricas, *WLAN* (Wireless Local Area Networks), son aquellas redes que hacen posible la interconexión de ordenadores en área local sin necesidad de cables. Es decir, su propósito es como el de las redes *LAN* clásicas pero sin cables.

Dichas redes pueden ser usadas con dos topologías distintas:

1. Redes ad-hoc: Son llamadas redes cerradas, son aquellas en las que los dispositivos se comunican directamente entre sí.
2. Redes de acceso inalámbricas: Los terminales se comunican con un punto de acceso (*AP*-Access Point), a través del cual pueden interconectarse y a su vez, acceder a redes externas.

Cabe destacar los siguientes protocolos de acceso estandarizados:

- *IrDa*, basadas en infrarrojos. Problema, necesita visión directa.
- *IEEE 802.11*: Familia de estándares adoptados por la *WiFi Alliance* para productos *WiFi*. Destacamos por su mayor implantación:
  - *IEEE 802.11a*: Se caracteriza por altas velocidades de hasta 54 Mbps en la banda de 5 GHz.
  - *IEEE 802.11b*: Es el estándar dominante en estas redes y adopta el nombre de *WiFi*. Soporta velocidades de 11 Mbps en la banda de 2.4GHz.
  - *IEEE 802.11g*: Ofrece velocidades de 54 Mbps en la banda de 2.4GHz, y es compatible con 802.11b.
- *HiperLAN2*: definido por la *ETSI* y que soporta velocidades de 54 MBps en la banda de 5GHz.

Debido a la alta adopción de los estándares *WiFi* en una gran mayoría de las redes inalámbricas locales instaladas en la actualidad va a ser el protocolo del que se hablará con más detalle en el siguiente apartado.

### 2.2.1 802.11

Las redes que se rigen por esta familia de protocolos son las denominadas *WiFi* [17]. *WiFi* significa Wireless Fidelity, que se corresponde con una certificación que otorga la organización *WiFi Alliance* a aquellos productos que superan unas pruebas de interoperabilidad para el estándar 802.11.

#### Características

- Siguen en líneas generales el *IEEE.802.3*, estándar de redes de área local.
- Usan *CSMA* como mecanismo de acceso al medio.
- Como banda de frecuencia usan la regulada en torno a los 2.4GHz, en Europa, y 5 GHz en EEUU, para aplicaciones industriales, científicas y médicas.
- Interfaz de tecnología radio:

Utiliza técnicas de modulación de espectro ensanchado (Spread Spectrum) [18]. El radiocanal empleado por estas redes tiene un ancho de banda de 22 MHz. Dichas técnicas emplean una baja densidad de potencia, por lo que la señal no interfiere con otros receptores y a su vez incorporan redundancia, por lo que proporciona a estos canales resistencia a interferencias y al ruido. Existen fundamentalmente varias técnicas:

- *FHSS* (Frequency Hopping Spread Spectrum):

El código ensanchado de los sistemas *FHSS*, es una secuencia de frecuencias específicas que se utiliza para modificar la frecuencia portadora de forma periódica cubriendo todo el canal asignado. La duración de cada salto está en el rango de los 100 ms. El mensaje se modula sobre la portadora utilizando *FSK*, lo cual genera una señal de banda estrecha para la duración de cada salto, pero de banda ancha si se observa el canal en intervalos de tiempo de varios segundos.

- *DSSS* (Direct Sequence Spread Spectrum):

La señal utiliza todo el ancho de banda del canal asignado, durante la duración de cada bit, la portadora se modula mediante *PSK*, siguiendo una secuencia de bits específica (chip). La portadora se modulará con un determinado chip en función de si el bit de mensaje aparece a cero o a uno, que se usará la misma secuencia pero invertida.

Si cada usuario utiliza un código chip diferente el canal podrá ser utilizado por varios usuarios de forma concurrente. Esta técnica se conoce con el nombre de *CDMA*, (Code Division Multiple Access).

En el 802.11, se utilizan códigos ortogonales de longitud 11, por lo que no es posible utilizar *CDMA*.

También se emplea otra técnica que no es de espectro ensanchado pero que proporciona un rendimiento bastante superior a la misma velocidad que las de espectro ensanchado. Esta técnica se conoce como *OFDM* (Orthogonal Frequency Division Multiplexing).

En *OFDM*, el ancho de banda asignado se divide en canales y la señal a enviar en tramas, de tal forma que cada una de ellas es modulada por la frecuencia de subportadora de cada canal. De esta forma se busca optimizar el canal de comunicaciones, reforzando los canales más débiles, aquellos más ruidosos o atenuados.

- Capa de acceso al medio:

*IEEE* 802.11 varia en su capa *MAC* con respecto al 802.3 para medios guiados, debido a que en medios no guiados como el aire, no es posible aplicar el mecanismo de detección de colisiones ya que el terminal no es capaz de realizar escuchas simultaneas. Por lo que se ha adoptado un protocolo para acceso al medio que se llama Distributed Coordination Function (*DCF*); es una variante de *CSMA* que se conoce como *CSMA/CA* (Carrier Sense Medium Access/ Collision Avoidance) con *ACK*.

Tiene dos modos de operación:

- Detección de canal físico:

Es el más básico de los dos. El emisor que desee transmitir una trama realiza una verificación previa del estado del radiocanal. En el caso de detectarse que dicho canal se encuentra inactivo, se comienza a transmitir inmediatamente. Si no, se queda a la espera y se transmite cuando se detecte que el medio queda liberado. Es necesario que el receptor de la trama confirme la llegada de la misma mediante un *ACK*.

Si hay colisión entre dos estaciones por haber transmitido al mismo tiempo, ambas esperan un breve tiempo aleatorio para volver a transmitir.

Es posible encontrarse con el problema conocido como “nodo oculto”. Como no todas las estaciones se encuentran dentro del alcance del resto, las transmisiones que se están produciendo en una parte de la red inalámbrica pueden no ser detectadas por el resto de los equipos conectados a dicha red.

- Detección de canal virtual:

También conocido como *RTS/CTS*. Cuando un terminal quiere transmitir a otro, se envía previamente una trama *RTS*, que contiene información de la trama y del emisor, la longitud de datos a enviar y la identidad del emisor. El terminal receptor responderá con una trama *CTS* en el caso de estar preparada para recibir la trama, donde especifica el emisor al que concede el canal y la longitud de datos a recibir.

Entonces el terminal emisor comienza a transmitir y se mantiene a la espera de un *ACK* del receptor para saber que los datos han llegado a su destino.

En este modo de operación se evita el problema del nodo oculto, pues los nodos que estén esperando si que escuchan el *CTS* con la información asociada, detectando así la ocupación del canal, con lo que sabe que durante el tiempo que transcurra hasta escuchar el *ACK* el canal está ocupado y por tanto no puede tratar de realizar una conexión con el receptor.

El problema en este caso viene derivado del inconveniente del esquema *CTS/RTS* que es la cantidad de información de control que se tienen que intercambiar entre si los nodos, siendo el volumen efectivo de intercambio de datos de usuario muy pobre en comparación en el modo de operación anterior.

El estándar 802.11 permite también efectuar control centralizado a través del protocolo *PCF* (Point Coordination Function), cuya implementación es opcional. En el *modo PCF*, una estación base consulta a las demás si tienen tramas que emitir mediante un sistema de sondeo o polling.

El mecanismo para establecer el sondeo se basa en la transmisión de una trama baliza (*beacon frame*) de manera periódica, entre 10 y 100 veces por segundo. A través de esta trama se indican las características del servicio de polling e invita a nuevas estaciones a conectarse al mismo. Una vez que una estación se adhiere a dicho servicio a una tasa de transmisión específica, se le garantiza una fracción del ancho de banda total de la estación base.

El *modo PCF* es la base para implementar calidad de servicio. *DCF* y *PCF* pueden convivir dentro de una misma red con lo que algunas estaciones poseerán calidad de servicio mientras que el resto competiría por el medio compartido. No obstante, en la actualidad el mecanismo de *QoS* no está prácticamente extendido ya que no todos los equipos *WiFi* están preparados para trabajar con el *modo PCF*.

## Arquitectura

Las redes inalámbricas pueden tener dos tipos de arquitectura totalmente distintos:

- Red Ad-Hoc:

También conocida como '*Peer-to-Peer*', Grupo de ordenadores que se comunican cada uno directamente con los otros a través de las señales de radio sin usar un punto de acceso. La red se conforma como un sistema distribuido en el cual ningún dispositivo controla al resto.

Pueden ser útiles para cubrir necesidades puntuales de comunicación, por ejemplo, comunicar los equipos durante una reunión o conferencia para ver la presentación, o pasarse archivos durante la misma.

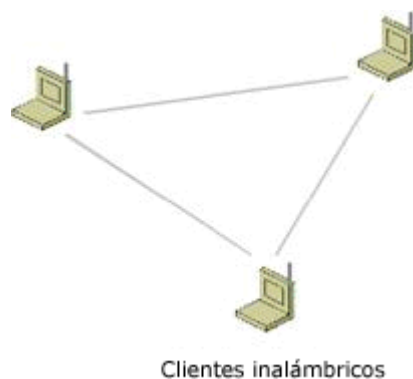


Fig 2: Ejemplo de Red Ad-Hoc.



- Red Modo Infraestructura:

Es como una especie de extensión de las redes locales cableadas. En éstas existe un dispositivo que es el que sirve para controlar al resto. Este esquema se configura mediante la utilización de los siguientes elementos:

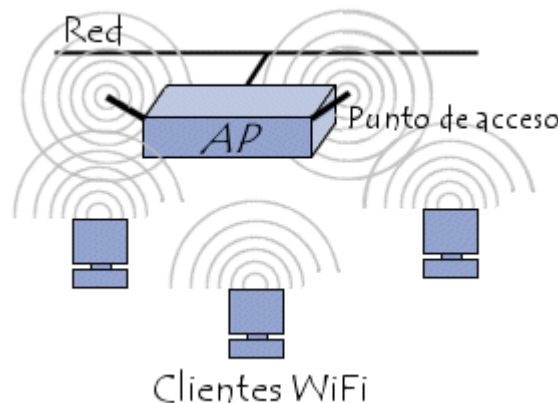


Fig 3: Ejemplo de Red en Modo Infraestructura.

- **Terminales de Usuario (Clientes):**

Requieren disponer de una tarjeta Interfaz de Red (*NIC*) y un transceptor radio.

- **Punto de Acceso (*AP*, Access Point):**

Control central de la red inalámbrica. Desde el punto de acceso se coordina la transmisión y recepción de múltiples dispositivos inalámbricos dentro de una ubicación específica; la extensión y el número de dispositivos dependen del estándar de conexión inalámbrica que se utilice y del hardware de los mismos.

Normalmente los puntos de acceso se encuentran conectados a un concentrador (hub) o a un conmutador (switch), y estos a su vez a un router *IP* para conectarse a la red.

- **Controlador de acceso (*AC*, Access Controller):**

Es un enrutador *IP*, que también suele funcionar como *AP*, y que se encarga de asignar direcciones *IP* a los terminales inalámbricos, mantiene una lista de direcciones de los terminales autenticados, y también filtra el tráfico, para descartar paquetes de terminales no autenticados.

Estos dispositivos sirven para montar una red inalámbrica independiente en la que se da cobertura a una zona grande mediante la utilización de entornos multicelda. Los *AP* de estas redes necesitan ser conectados a un controlador de acceso para que la red sea totalmente independiente.

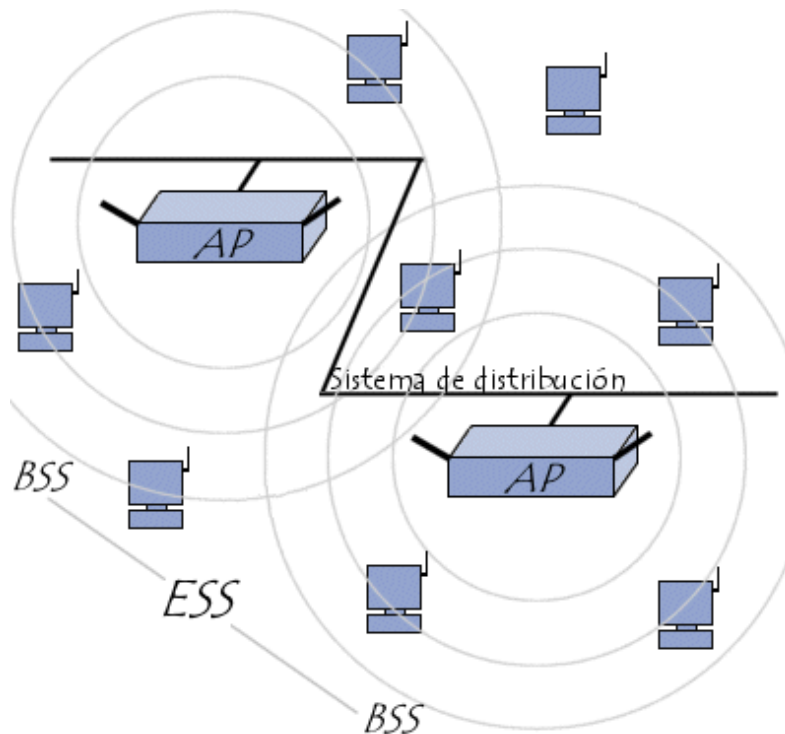


Fig 4: Ejemplo de sistema con Controlador de Acceso.

Para aumentar la capacidad de la red inalámbrica, se puede disponer más de un AP iluminando la misma zona con distintos canales de frecuencia. Por limitaciones del ancho de banda disponible, el número de AP permitidos en una misma zona depende de la banda de trabajo a la que se actúe. Además, para evitar interferencias entre ellos, los canales tienen que estar separados 22MHz. Todo esto provoca que para redes que trabajen en la banda de 2,4GHz sólo puedan operar sin interferencias 3 AP concurrentes, mientras que en la banda de 5 GHz se puede disponer de 8 AP.

## Seguridad

Debido a las frecuencias de radio en las que operan las redes inalámbricas es fácil que un tercero pueda intentar el acceso a la misma sin autorización ninguna. Puesto que las señales no pueden ser confinadas dentro de los muros de un recinto. Por lo tanto es necesario definir mecanismos de seguridad para evitar estos problemas. Debido a la naturaleza de las mismas es necesario que sean mecanismos estrictos, para autenticación en red o de criptografía para proteger las tramas ante escuchas. Los mecanismos de seguridad a implementar suelen definirse en función del tipo de arquitectura en el que este funcionando la red.

En topología ad-hoc suelen establecerse dos posibles mecanismos para conformar la red y permitir el acceso a la misma:

- **Open System:**  
Lista de usuarios definidos, de manera que cualquier terminal cliente puede solicitar la autenticación, y el terminal que la recibe la otorgará sólo a las estaciones que se encuentran en su lista de usuarios.
- **Shared Key:**  
Solamente las estaciones que comparten una clave secreta pueden ser autenticadas

En el modo infraestructura la autenticación se otorga mediante un diálogo cifrado entre el cliente y el AP. Los mecanismos que se usan son:

- **WEP (Wired Equivalent Privacy):**  
  
Este protocolo cifra la información mediante el algoritmo simétrico *RC4* con claves de 40, 128 o 256 bits. Las claves no se intercambian por el canal aire, para evitar la interceptación.  
  
Tiene vulnerabilidades conocidas que permiten descifrar la información en muy poco tiempo.
- **WPA (WiFi Protection Access)**  
  
Es una solución más robusta que la anterior que ha sido desarrollada dentro del grupo IEEE 802.11. Se basa en el intercambio de claves temporales utilizando el algoritmo simétrico Rijnjael (AES- Advanced Encryption Estandar). WPA es el producto comercial de esta solución.
- **ACL (Access Client List)**  
  
Validaciones *MAC* de los clientes contra unas listas de acceso.
- **VPN**  
  
Configuración de redes de acceso privadas virtuales.

Algunos de los mecanismos de seguridad que se pueden implementar son comunes a cualquier red de la familia 802, sea inalámbrica o no. En este sentido se ha aprobado el

estándar 802.1x, en el cual se establece un control basado en puertos, de manera que el puerto que se le asigne a un terminal se mantendrá inhabilitado hasta que el servidor lo valide por medio del protocolo *EAP* (Extensible Authentication Protocol) contra un Servidor de Autenticación (*RADIUS*).

### 2.2.2 Ventajas e inconvenientes

Las redes inalámbricas de área local, en su vertiente *WiFi*, han alcanzado un gran éxito comercial, y su despliegue en el mercado residencial se ha visto favorecido por su utilización como técnica de acceso inalámbrico en el hogar a la conexión vía *ADSL* a Internet. En las organizaciones también están experimentando un alto nivel de penetración. Esto viene motivado fundamentalmente por las ventajas que ofrece esta tecnología frente a otras alternativas:

- Movilidad de los usuarios.
- Sencillez de instalación.
- Flexibilidad en la elección de ubicaciones.
- Permite su dimensionamiento en función del tráfico y preferencias de los usuarios.
- Robustez y bajo mantenimiento: no es necesario mantener cableado con lo que se evitan los problemas asociados.
- Escalabilidad: permite crecer de forma natural y requiere poca planificación, ya que añadir nuevos puntos es inmediato, Es fácil extender la red y cubrir nuevas zonas, y permite añadir nuevos usuarios sin necesidad de nuevos puestos físicos.

No obstante, las redes inalámbricas presentan a su vez una serie de inconvenientes que frenan su generalización e impiden que se puedan plantear como alternativa real a las redes ya existentes de cable. Entre otros podemos indicar:

- Aumento del riesgo en la seguridad, al no tener contenidas las comunicaciones en un medio físico guiado.
- Alcance máximo inferior al de las redes cableadas.
- Menor velocidad que las cableadas y tasa de error mayor, lo cual conlleva la retransmisión de paquetes y por tanto la disminución de la tasa efectiva de bit.
- Compartición de la capacidad de transmisión con el resto de usuarios de la red, lo que limita el número de puestos para poder mantener un nivel de servicio dado.
- Regulación del espacio radioeléctrico, que limita las frecuencias a utilizar y por tanto restringe la capacidad máxima de los sistemas.

## **2.3 Modelos de sistemas de localización**

Los sistemas más usados en posicionamiento se pueden clasificar en dos grandes grupos [1] :

- Aquellos en los que se hace uso de conocimiento ‘a priori’, apoyándose en el uso de técnicas de minería de datos, a los que se denominan “Empíricos”.
- Y por otro lado, aquello que se basan únicamente en modelos de propagación de radiofrecuencia, con procesamiento de señal, a los que se denominan “de Propagación”.

### **2.3.1 Modelos empíricos**

Los modelos empíricos son aquellos en los que el sistema de posicionamiento a construir se basa en la recolección de información del entorno ‘a priori’. Dicha información debe ser clasificada y almacenada en base de datos, siguiendo modelos de “minería de datos”.

A partir de dicha información se puede obtener la posición de futuros elementos en la red, con distintas técnicas:

- distancia mínima,
- k-vecinos más próximos,
- redes neuronales
- y un largo etc.

Los pasos a seguir en estos sistemas son:

1. Conocimiento del entorno.

Se debe definir el entorno en el que se quiere implementar el sistema, por ejemplo, definiendo el mapa de la localización en el que se va a mover y los puntos de acceso que existen en la red existente (BSAs, Basic Service Areas); así como las posibles zonas o puntos que se quieran definir en el sistema.

2. Calibración del entorno.

En este punto se debe recolectar la información de la red en la que se están moviendo los terminales. Por ejemplo, tomando medidas de la señal y almacenándolas en una base de datos correctamente clasificadas. Para poder calibrar el sistema se debe conocer bien el estándar 802.11 [1], para saber los datos de los que se puede hacer uso.

### 3. Posicionamiento de nuevos elementos.

Será el último paso, en el cual cuando se introduzca un elemento nuevo en la red, se posicionará basándose en la información que se conoce del sistema.

Este modelo de sistemas de posicionamiento requiere un gran esfuerzo ‘a priori’ para modelar y obtener la información del sistema en el que se va a mover; así como un buen mantenimiento de los datos que calibran el sistema, debido a que las redes inalámbricas de interior son redes muy variables, en las que las señales fluctúan mucho debido a la presencia de objetos, personas, introducción de nuevos sistemas inalámbricos, etc.

#### ***2.3.2 Modelos de Propagación***

Los modelos de propagación, como bien se puede intuir por su nombre, se basan en los sistemas de radioenlaces y las técnicas de multitrayecto de la señal de radiofrecuencia. La posición se obtiene usando como referencia los puntos de acceso más cercanos. Uno de los métodos más usados en este caso, es el de triangulación.

Este modelo suele ser muy preciso, pero muy lento en el cálculo y la obtención de información del medio, a veces innecesaria para el cálculo en el modelo elegido.

### ***2.4 Principales métodos de posicionamiento en sistemas de localización WiFi.***

#### ***2.4.1 Vector de Potencias***

Este método [24] se usa en aquellos modelos de sistemas empíricos, usando la técnica de “distancia mínima”.

Para usar este método en primer lugar es necesario conocer los *AP*’s que se disponen y su información. Para entrenar el sistema se almacena en una base de datos las señales obtenidas desde los distintos *AP*’s por el usuario en distintos puntos; para poder estimar la posición del usuario posteriormente buscando los niveles de señal más cercanas al vector de señales que se obtengan en esos momentos, comparando los vectores de potencia con una medida de distancias, por ejemplo, la distancia euclídea.

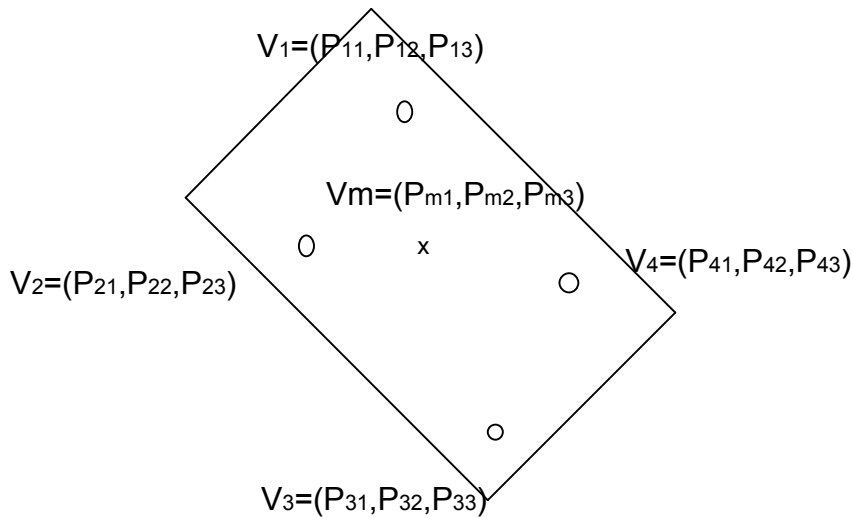


Fig 5: Ejemplo de Cálculo de Método Vector Potencia.

### 2.4.2 Triangulación

Es el proceso [30] mediante el que se determina la ubicación de un punto mediante la medición de los ángulos a puntos conocidos, en lugar de medir las distancias al punto directamente. El punto a medir puede fijarse con el tercer punto de un triángulo con 2 ángulos conocidos y un lado.

La triangulación mediante consiste en averiguar el ángulo de cada una de las tres señales respecto al punto de medición. Conocidos los tres ángulos se determina fácilmente la propia posición relativa respecto a los tres puntos de referencia. Conociendo además las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtiene la posición absoluta o coordenadas reales del punto de medición.

### 2.4.3 Trilateración

La trilateración [30] es un método de localización, que necesita conocer dos o más puntos de referencia, así como la distancia media entre el usuario a localizar y cada uno de dichos puntos. Para poder determinar de forma única y precisa la localización relativa de un punto en un plano bidimensional usando sólo trilateración, se necesitan generalmente al menos 3 puntos de referencia, para tener en cuenta los problemas de precisión debido al tiempo necesitaremos 4 puntos de referencia.

El concepto de trilateración se ha adoptado en sistemas de localización de interiores, debido al gran éxito que está obteniendo como método de posicionamiento en el sistema GPS.

El receptor detecta un primer punto de acceso y determina que debe de estar a una distancia  $R1$  del mismo. Se conoce donde se ubica dicho  $AP$  y a su vez se sabe que el receptor debe estar ubicado dentro de la superficie de una esfera de radio  $R1$ .

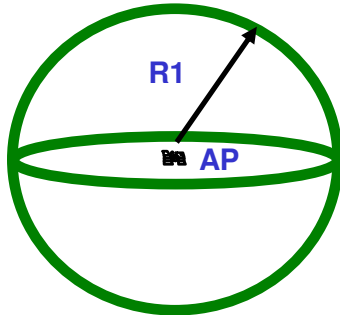


Fig. 6: Trilateración–Detección de 1 AP.

Al detectar el segundo  $AP$ , se realiza el mismo procedimiento por el cual se decide que el receptor debe estar sobre la superficie de una esfera de radio  $R2$ .

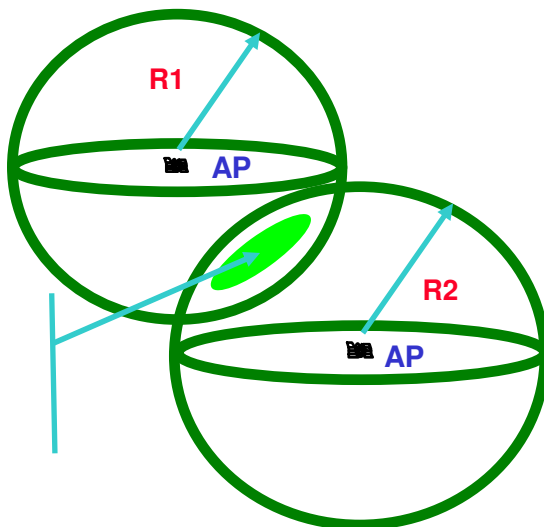


Fig 7: Trilateración- Detección de 2 APs.

La intersección con la primera esfera forma un círculo en cuyo perímetro puede estar ubicado el usuario.

Al realizar la medida con un tercer  $AP$  nos acerca a dos puntos únicos resultantes de la intersección de las tres esferas.



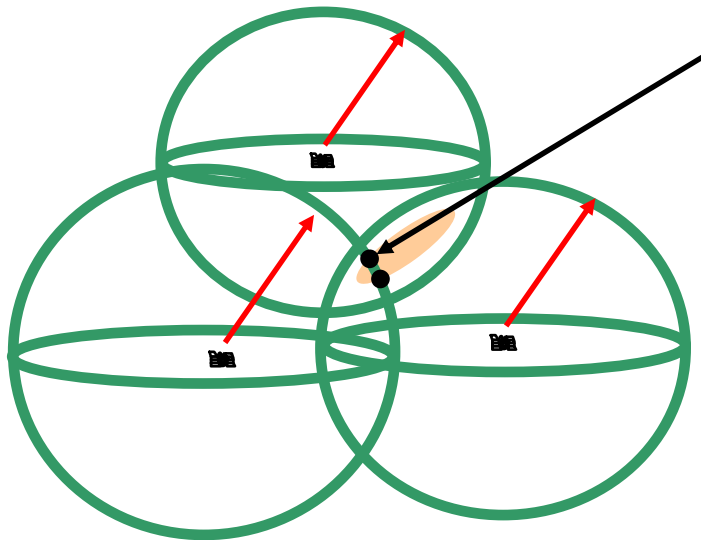


Fig 8: Trilateración- Detección de 3 APs.

Uno de estos dos puntos puede ser descartado por ser una respuesta incorrecta, bien sea por estar fuera del espacio por moverse a una velocidad muy elevada. Por lo que se necesita una cuarta medición a otro satélite para calcular las cuatro incógnitas,  $x, y, z$  y tiempo.

#### 2.4.4 Técnicas heurísticas de optimización.

Es una capacidad de innovación de los sistemas, basada en el aprendizaje y la experiencia. Se pueden utilizar por si solos o como mejora de los dos métodos principales, descartando posiciones de la base de datos para reducir las posibles localizaciones. Ejemplos característicos:

1. Proximidad

Se busca el Punto de acceso del que estamos recibiendo mayor potencia de señal, consideramos a dicho punto de acceso el más cercano y por lo tanto, nuestra posición será la de dicho punto de acceso.

2. K-vecinos más próximos

Es la técnica más conocida y usada. Es una técnica de aprendizaje que consiste en almacenar una serie de ejemplos en la fase de entrenamiento, cuando intentamos localizar una nueva posición consultamos los datos de entrenamiento y ubicamos al cliente por proximidad. Se requiere un sistema suficientemente entrenado para poder localizar con precisión.

### 3. Redes neuronales

Son sistemas capaces de aprender y adaptarse a diversas condiciones muy rápidamente. Igual que el k-vecinos más próximos necesita una fuerte fase de entrenamiento.

### 4. Teoría de Bayes

Técnica probabilística que mantiene una distribución de probabilidad sobre todas las posibles ubicaciones del entorno. Las técnicas probabilísticas consiguen una precisión superior que las técnicas deterministas a cambio de un mayor coste computacional. La aproximación Bayesiana se suele aplicar en los casos en los que la representación del entorno es en forma de rejillas. Otra alternativa para modelar el entorno es mediante un mapa topológico. En este caso la localización se basa en el hecho de que el dispositivo identifica automáticamente que ha alcanzado un nodo del mapa en base a alguna información geométrica del entorno.

### 5. Movimiento

Es posible clasificar a un usuario como parado o en movimiento basándose en la fuerza de la señal *WiFi*. Hay que tener en cuenta que la potencia de la señal de los *AP*'s hace más picos alrededor de la posición estimada cuando el dispositivo está en movimiento que cuando está parado. Por lo que hay que entrenar el dispositivo en movimiento y parado alrededor de una zona.

## ***2.5 Proyectos de Sistemas de Localización y Sistemas Comerciales.***

En estos últimos años son muchos los estudios y sistemas llevados a cabo sobre sistemas de posicionamiento en redes inalámbricas, se puede encontrar bastante literatura al respecto así como algunos sistemas comerciales implantados con mejor o peor resultado.

Se intentará hacer un resumen del estado del arte de los sistemas y estudios que han servido de apoyo para realizar el diseño de este proyecto.

### 2.5.1 RADAR

*RADAR* [22] es uno de los proyectos pioneros en el diseño de sistemas de posicionamiento/localización en redes inalámbricas en interiores. Fue promocionado por Microsoft Research en el año 2000, por lo que se utilizaron ordenadores bajo Windows 95 y estaciones base con *FreeBSD* 3.0.

El proyecto *RADAR*, modela su sistema haciendo una mezcla de los 2 modelos presentados anteriormente, el ‘empírico’ y los ‘modelos de propagación’; haciendo uso en estos últimos tanto del nivel de señal recibida como de la relación señal a ruido existente.

En primer lugar recogerá datos de la señal y su relación señal a ruido, creando un mapa de señal de las zonas, para posteriormente poder validar/caracterizar el modelo de propagación a utilizar para estimar la posición. El modelo de propagación utilizado es el de la ecuación del Factor de Atenuación de Muros, *WAF*, para calcular las distancias a partir de las pérdidas del trayecto.

### 2.5.2 SkyhookWireless

*Skyhook* [28] nace en el año 2003 por la influencia de la intensa demanda de servicios de localización tanto en interiores como en exteriores. La idea comienza con un sistema híbrido de localización mediante *GPS* y algoritmos de triangulación.

En vista de los problemas de cobertura y el tiempo invertido en recuperación, nace bajo la mano de *Skyhook*, el sistema *WPS*, “WiFi Positioning System”. Es un sistema híbrido que aprovecha las ventajas de la gran red *WiFi* existente en una gran número de ciudades, para aprovechando estos beneficios, junto con los métodos de triangulación utilizados en *GPS* para desarrollar un sistema ‘indoor’, que proporciona información de localización muy precisa tanto en interiores como en zonas de población urbana densa.

Es un sistema que proporciona la misma calidad tanto en interiores como en exteriores. Este sistema se está utilizando en la actualidad en la *CNN*.

Características:

- Usa triangulación en sus cálculos, así como un modelo empírico.
- Funciona tanto en entornos ‘indoor’ como ‘outdoor’.
- **Velocidad:**  
1 segundo en búsqueda de localización
- **Precisión:**  
10-20 metros, en interiores y exteriores
- **Fiable:**  
En cualquier lugar.

### 2.5.3 *Amulet*

*Amulet* [3] (Approximate Mobile User Location Tracking System), es un sistema de posicionamiento en entorno *Linux* para interiores desarrollado en la Universidad de Rochester basado en la infraestructura de *IEEE 802.11 WLAN*.

Usa un modelo mixto muy similar al proyecto *RADAR*, realiza un ‘radio-map’ almacenando muestras de la señal en una base de datos, para posteriormente buscar el más cercano mediante el algoritmo del k-vecinos más próximos. Obtiene una precisión peor a la de *RADAR* de en torno a 3-5 m de media, debido a la debilidad del algoritmo de k-vecinos más próximos, ya que no tiene en cuenta ningún algoritmo de seguimiento.

Principales características:

- Sistema mixto, modelo de propagación y empírico.
- Dispone de una fase de entrenamiento en la que construye un ‘radio map’ con las señales *Wifi* existentes y las almacena en una base de datos. Y una fase posterior de cálculo de localización mediante el algoritmo de los k-vecinos más próximos.
- Precisión: 3-5 metros en media.

### 2.5.4 *Horus*

*Horus* [25] es un sistema de localización *RF* basado en la infraestructura de las redes inalámbricas *IEEE 802.11*. Implementa una versión probabilística de un modelo matemático de propagación de señal.

El ‘radio-map’ almacena las distribuciones de señal que tienen cada punto de acceso en cada localización. Dichas distribuciones se almacenan tanto con funciones paramétricas como con no paramétricas.

La versión paramétrica usa distribuciones Gaussianas sobre las que aplica la función autorregresiva de primer orden para describir la fuerza de la señal 802,11, suponiendo independencia de ruido.

### 2.5.5 *Halibut*

*Halibut* es un proyecto de la Universidad de Stanford, el cual usa exclusivamente un modelo de propagación para el posicionamiento. Los parámetros que se tienen en cuenta en el modelo de radio-propagación son pérdidas en espacio libre, atenuación de señal, difracción, desvanecimiento por multitrayecto, y variable aleatoria de tipo log-normal.

Principales características:

- Usa únicamente un modelo de propagación, teniendo en cuenta multitrayecto y desvanecimiento.
- Precisión variable en función del entorno.
- Lentitud debido a la cantidad de cálculos a realizar en tiempo real.

### **2.5.6 LEASE**

*LEASE* [14], “Location Estimation Assisted by Stationary Emitters”, es decir es un sistema de estimación de localización asistido por emisores estacionarios para redes inalámbricas en interiores. Este sistema funciona sobre entornos Linux.

Principales características:

- Localización sólo en interiores.
- Usa técnicas no-paramétricas para la localización, vectores de potencia y K-vecinos más próximos.

Su arquitectura se basa principalmente en 3 elementos:

- ‘Stationary Emitters’ (*SE*).

Son emisores inalámbricos 802.11 con posición fija y conocida, que transmiten unos pocos paquetes a la hora que serán reconocidos por los sniffers. Estos dispositivos no necesitan conexión ninguna a la red.

- Sniffers

Están escaneando el entorno inalámbrico de forma periódica para recoger las señales recibidas tanto por los *SE* como por los clientes. Se encargan de transmitir dicha información al *LEE*.

- ‘Location Estimation Engine’ (*LEE*).

Mantiene una tabla con la localización (x,y) y la identificación (*MAC* id) de los *SEs*. Con la información transmitida por los sniffers y la tabla que posee de los *SE* construye un modelo para poder estimar la localización de los clientes. Por lo que genera un mapa de vectores de potencia y estima las posiciones mediante un *K-NN* (K-Vecinos más próximos).

### 2.5.7 Ekahau Positioning Engine

*Ekahau* [6] es una herramienta de posicionamiento en redes inalámbricas 802.11, es un sistema mixto entre el modelo empírico y el de propagación. Es el único software del mercado de sistemas de localización en tiempo real sobre multitud de plataformas, desktop, portátiles, *PDA*... Funciona tanto en interiores como en exteriores, únicamente se necesita cobertura en una red local inalámbrica (*WLAN*).

Sus principales características son:

- Es compatible tanto con dispositivos de *Ekahau*, pasivos, etiquetas o Tags T201 como con dispositivos activos, ordenadores, *PDA*'s...
- Localiza varios dispositivos simultáneamente sobre un mismo mapa de situación.
- La información que proporciona de cada dispositivo son, coordenadas (x,y), edificio, piso, habitación y zona.
- Funciona sobre cualquier 802.11.
- Cada usuario debe instalarse el software cliente y calibrar los mapas de área para poder comenzar a usar el sistema.
- Su precisión es de hasta 1 metro.

El sistema *Ekahau* [7] funciona mediante el uso de tres módulos:

- *EPE* (Ekahau Positioning Engine)

Es un servidor de aplicación, que se usa como Centro de Control, calcula las coordenadas de los dispositivos cliente y almacena los datos de calibración.

- *EM* (Ekahau Manager)

Es la aplicación que se encarga de la calibración de los dispositivos y de las áreas, así como de "live tracking".

- *EC* (Ekahau Client)

Este software debe ser instalado en todos los dispositivos cliente, es el responsable de recoger el nivel de potencia de señal y toda aquella información que recoge la tarjeta de red inalámbrica del mismo, y enviarla.

*Ekahau* trabaja en dos fases, una de entrenamiento y otra de cálculo de posicionamiento. En la fase de entrenamiento los *EC*, miden las señales y envían dicha información a los *EM*, para que creen el vector de potencias. Para posteriormente realizar el cálculo de la posición con la técnica de *WiFi Mapping*.

### 2.5.8 Herecast

*Herecast* [11] es un sistema que proporciona servicios de localización basados en *WiFi* (802.11). Se basa en los modelos empíricos, usando una base de datos con la posición de todos los puntos de acceso y un algoritmo para calcular la ubicación del usuario basándose en esos datos.

Sus principales características son:

- Los cálculos de posicionamiento se realizan sobre el dispositivo, por lo que no es necesario rastrear al cliente desde el servidor central.
- Utiliza un sistema simbólico para expresar la localización, de forma que los usuarios sepan más fácilmente donde están sin tener que traducir las coordenadas.
- Su infraestructura es accesible por todos los usuarios.
- Ofrece una buena resolución con un único Punto de Acceso en una sala grande. La posición que estima el sistema no son unas coordenadas concretas sino un área y unos datos que el usuario define en el entrenamiento y se guardan en la base de datos.
- De cada punto de acceso se debe saber: país, provincia, ciudad, área, edificio y planta.
- La base de datos es dinámica, se va modificando cuando un usuario descubre información de un nuevo punto de acceso, dicha información será accesible por cualquiera de los usuarios.

El sistema usa tres elementos en su arquitectura:

- Los puntos de acceso

*Herecast* relaciona los códigos de identificación de los puntos de acceso con un texto entendible por el usuario que marca la localización donde se encuentra, al cual llama “landmark”.

- La base de datos

En ella se almacena toda la información referente al punto de acceso, su código e información de zona. Cuando un usuario se conecta a un *AP*, el sistema busca dicho código en la base de datos y muestra la información de la zona. Está puede ser actualizada por cualquier cliente del sistema.

- Clientes *Herecast*

Es el software que se instala en los dispositivos para permitir su localización y diversas aplicaciones asociadas a ella.

*Herecast* utiliza un sistema de nombramiento simbólico para reconocer un punto de acceso cercano, y mediante una heurística de proximidad, como método principal de localización, sabemos la posición en la que se encuentra.

## 2.5.9 *Placelab*

*Placelab* [20] es un software para aplicaciones de posicionamiento tanto en interiores como en exteriores. Fue desarrollado por investigadores de la Universidad de Washington, de California en Berkeley y por Intel en Octubre 2004. Esta desarrollado en J2ME y actualmente se puede correr tanto sobre Windows XP, como Linux, Pocket PC 2003, y Symbian.

Sus principales características son:

- El cálculo de posicionamiento se realiza desde el lado del cliente, por lo que no necesita estar en constante contacto con el servidor central.
- Los dispositivos se posicionan basándose en el monitoreo pasivo del entorno.
- Permite localización mediante, *WiFi*, *GSM* y dispositivos *Bluetooth* y *GPS*. Por ello, se puede tanto localizar dispositivos en ambientes “indoor” como “outdoor”, al contrario que *GPS* que solo localiza bien en entornos “outdoor”.
- Su precisión media teórica es de entre 15 y 20 metros, o el doble del error que hay en *GPS*. Su cobertura y exactitud dependen del número y el tipo de puntos de acceso que haya alrededor del dispositivo del cliente.

La arquitectura de *Placelab* se basa en tres elementos distintos:

- Radio beacons

Se llaman así a los puntos de acceso del entorno del dispositivo móvil. Pueden ser *WLAN* Access Points, dispositivos *Bluetooth* fijos, torres celdas *GSM*. Cada uno de ellos se reconoce por un identificador único, que es su dirección *MAC*. Mediante ese identificador el cliente calculará su posición más fácilmente. Estos Puntos de Acceso no tienen porque ser de la propia red, sirven todos aquellos que estén a la vista.



- Base de Datos

Mantiene la información de los “radio beacons”, sus coordenadas y su identificador, *MAC*. Se puede usar una base de datos local o una base de datos remota a la que tengan acceso todos los usuarios a localizar.

- Clientes

Es el software instalado en el dispositivo del cliente, tiene 3 modos de funcionamiento:

1. “Spotters”,

Cada cliente instancia un “spotter” por protocolo de acceso soportado por el dispositivo; así se puede tener un “spotter” para Bluetooth y otro por *WLAN*. Su tarea es monitorizar la interfaz radio y compartir los identificadores observados de los “radio beacons”.

2. “Mappers”,

Proporciona la localización de los “radio beacons” conocidos, puede obtener la información directamente de la base de datos o de una base de datos remota compartida.

3. “Trackers”,

Estima la posición del usuario usando los datos obtenidos previamente por el Spotter y el Mapper.

El funcionamiento del sistema *Placelab* está basado en los modelos empíricos. Primero se añade una base de datos con la coordenadas de los Puntos de Acceso conocidos y su dirección *MAC*, para así quedar reconocidos y poder identificar los demás Puntos de acceso vistos por el cliente como desconocidos.

*PlaceLab* no tiene fase de entrenamiento, se realiza la fase de reconocimiento y de estimación de localización en los propios dispositivos. Los trackers realizan la estimación basándose en métodos heurísticos, como el método de Centroides, dichos métodos pueden ser elegidos en el sistema eligiendo distintos trackers.

## 2.5.10 Google Latitude

Google Latitude [10] es un servicio de localización de personas en tiempo real. Está basado en la tecnología desarrollada por Dodgeball, empresa posteriormente adquirida por el buscador californiano, y que una de las compañías pioneras (fuera de Japón) en los servicios basados en localización para masas.

Google Latitude permite a un usuario compartir con sus contactos (almacenados en Google Mail) su localización, así como ver la localización de estos. Cada usuario puede escoger entre actualizar su localización automática o manualmente, así como el mecanismo para obtener su localización actual.

La aplicación reside en un teléfono móvil compatible o en un ordenador personal, y se comunica con otros servicios de Google para ofrecer posibilidades adicionales: a través de Google Maps se dibuja un mapa con la localización actual del usuario y la de sus contactos, siendo posible comenzar llamadas de voz o conversaciones de texto a través de Google Talk.

Cada usuario, en cualquier momento, puede decidir si desea continuar mostrando su localización, disminuir la precisión con la que su localización se muestra (indicando, por ejemplo, sólo la ciudad en la que se encuentra), o ocultarla totalmente.

Google Latitude comenzó a funcionar durante los primeros meses de 2009, por lo que no todas las posibilidades prácticas de los servicios basados en localización son explotadas actualmente. Dada la posición dominante de Google en la publicidad on-line, es especulación común en todos los medios tecnológicos que Google comenzará a incluir publicidad basada en la localización a sus usuarios.

Google Latitude cuenta con tres mecanismos distintos para determinar la posición actual de un usuario:

- *Identificador de celda:*

Si el dispositivo móvil del usuario lo permite, Google puede leer el identificador de la celda a la que el móvil se halla conectado.

- *GPS:*

Google es capaz de interactuar, en los dispositivos adecuados, con el GPS que algunos móviles de gama alta (como el dispositivo Q9 de Motorola) comienzan a ofrecer.

- *WiFi:*

Si el móvil (o un ordenador personal) está conectado a una red WiFi a través de un punto de acceso identificable, y Google posee información sobre ese punto de acceso, podrá utilizarse dicha información de forma muy parecida al identificador de celda.

Evidentemente, la precisión con la que Google Latitude es capaz de determinar la posición actual del usuario depende mucho tanto del mecanismo disponible para detectarla como de condiciones externas (por ejemplo, en un área rural las celdas de las

redes móviles son mucho mayores que en ciudad, lo que disminuye la precisión de un sistema de localización que sólo conozca el identificador de celda y asuma que el usuario no puede estar muy lejos de la estación base, cuya posición *GPS* se conoce)

Para el caso de localización a través del identificador de celda, hay que tener en cuenta que una celda típica en una red *GSM* puede tener entre 2km de diámetro (en entornos urbanos) y 20km (en entornos rurales). Sin embargo, existe información adicional a la que Google Latitude puede acceder si existe un acuerdo entre el buscador californiano y el proveedor de servicio telefónico involucrado, y que permitiría aumentar significativamente la precisión: el sector o subcelda y el “timing advance”.

Las celdas se hayan divididas habitualmente en sectores o subceldas, con lo cual, obtener el identificador limita la posición del móvil a una parte del área total dominada por la celda. Adicionalmente, la información de temporización intercambiada entre el móvil y la estación base permite estimar de forma un poco burda la distancia entre el móvil y la estación base, lo que limita la posición del móvil a una zona determinada de la subcelda.

No existe, sin embargo, información alguna que permita afirmar que Google ha firmado un acuerdo con los proveedores de telefonía móvil, por lo que, a día de hoy, su servicio de localización, en el caso de no disponer de información *GPS* o *WiFi*, ha de basarse, exclusivamente, en la localización de la estación base, con la falta de precisión a ello inherente.

Un caso muy distinto, es, sin embargo, el soporte de *GPS* que algunos móviles y computadores personales de última generación incluyen. En este caso, la posición del usuario puede determinarse con tanta precisión como lo permita el sistema *GPS*, lo que depende, entre muchos otros factores, de la calidad del receptor, la cantidad de satélites visibles en ese momento, los obstáculos a la señal, etc.

Adicionalmente, para aquellos móviles u ordenadores personales que usen *WiFi*, es posible comunicar información acerca del punto de conexión, por si se conociera la posición *GPS* de dicho punto. Esto es similar a la localización mediante identificación de celda, pero bastante más precisa, ya que la cobertura de un punto de acceso *WiFi* suele ser muy inferior a la de una celda en *GSM*. Sin embargo, aunque existen desde hace tiempo muchos servicios que cuentan con una base de datos acerca de las celdas de los operadores de telefonía móvil y su posición geográfica <sup>1</sup>, las bases de datos con posiciones de las estaciones base son relativamente recientes y no se han realizado estudios sobre su precisión o el porcentaje de puntos de acceso *WiFi* para los que se posee información.

Google Latitude soporta una gama creciente de teléfonos móviles. En este momento (Marzo 2009), un móvil que tenga instalado el sistema operativo Android, Windows Mobile 5.0 o superior, Symbian 3G o superior, Palm OS 5 o superior, pueden instalar la

---

<sup>1</sup> para un ejemplo gratuito, véase la API ofrecida por el rival de Google, Yahoo, para obtener la posición de una estación base conociendo su identificador de Celda.

<http://developer.yahoo.com/yrb/zonetag/locatecell.html>



aplicación de Google Latitude. Además, los dispositivos BlackBerry de *RIM* y muchos móviles basados en Java pueden instalar también la aplicación.

Sin embargo, no todos los dispositivos ofrecen una *API* adecuada para el acceso al identificador de celda o a su información de *GPS*. Google mantiene actualizada en su página web una completa lista de móviles soportados y los mecanismos de localización disponibles para estos, aunque ahora mismo se puede observar un claro predominio de los terminales disponibles para los operadores de EE.UU. y de los socios de Google

## ***3 Estudio de Viabilidad del Sistema***

### ***3.1 Descripción del Sistema***

#### ***3.1.1 Ámbito***

En el capítulo anterior del Estado del Arte se han visto distintos algoritmos y sistemas de localización existentes. Se puede apreciar que cada uno de los sistemas expuestos se basa en un único método de localización más o menos complejo, con ciertas ventajas o desventajas en cuanto a rapidez o eficiencia en función del sistema escogido empírico, de propagación o híbrido.

La razón principal del desarrollo de este sistema de localización es aunar las ideas concebidas por el estudio de los sistemas anteriores, desarrollando un sistema completamente modular que sirva para poder evaluar distintos algoritmos de localización sobre un mismo sistema.

El ámbito de este proyecto son las redes locales inalámbricas 802.11, el diseño se realizará teniendo en cuenta que éste es el entorno en el que se va a implementar el sistema.

#### ***3.1.2 Descripción General***

En esta sección se presenta una descripción de las características que se desean en el sistema de localización a realizar, sin tener en cuenta los medios para realizarlo.

##### ***3.1.2.1 Funciones del sistema***

El sistema debe incluir las siguientes características:

- Gestión de la Localización.
- Gestión de los usuarios.
- Calibración de la Localización.
- Monitorización del Sistema.
- Determinación de la Localización
- Presentación de la Localización

- Gestión de la Localización.

Para localizar a un usuario y que éste sepa con certeza dónde está en cada momento el sistema de localización necesita un mapa de la planta. El mapa deberá estar dividido claramente en zonas mediante coordenadas. Las coordenadas delimitarán un rectángulo mediante coordenadas  $(x_a, y_a, x_b, y_b)$ , midiéndose en píxeles y siendo el origen de coordenadas la esquina superior izquierda.

El mapa será en 2D, y en él no habrá posibilidad de incluir más de una altura. Para mejoras futuras se prevé incluir varias alturas de un edificio.

Se debe almacenar el mapa, las coordenadas de las zonas delimitadas y el nombre de la localización.

- Gestión de los Usuarios.

Hay tres tipos de usuarios, cada uno de los cuales con unas preferencias distintas. El primer tipo de usuario es aquél que quiere localizarse o ser localizado dentro de una planta, el segundo tipo es aquel que gestiona el sistema y hace que sea posible la localización, y por último tendremos el usuario que únicamente desea conocer la posición de los dispositivos móviles en la localización.

- Calibración de la Localización.

Para permitir la localización a los usuarios, el sistema ha de obtener los datos necesarios para determinar la zona donde se sitúa. Para ello hay que obtener la potencia de la señal recibida de todos los puntos de acceso conocidos en cada una de las zonas, para posteriormente hallar su localización más probable dado un conjunto de potencias de señal recibida.

- Monitorización del Sistema.

Visualización de la posición de todos los usuarios registrados que se encuentran en la localización durante los 30 min anteriores a la petición. Para ello necesitará, el plano de la planta, las coordenadas de las zonas y la posición en la que se encuentran los usuarios. El usuario mandará la localización de la cual quiere saber el posicionamiento de todos sus usuarios. El sistema calculará la posición de cada uno de los usuarios mediante un algoritmo de localización y los datos proporcionados por cada uno de los usuarios anteriormente.

- Determinación Presentación de la Localización.

El usuario pedirá todos los datos necesarios al sistema, previa conexión y autenticación, para poder saber su localización. Para ello necesitará, el plano de la planta, las coordenadas de las zonas y el nombre de la localización en la que se encuentra. Mandará la potencia de la señal recibida de los puntos de acceso captados en ese momento y el sistema le devolverá la zona en la que se encuentra. El sistema calculará la zona usando el método de vector de potencia como ejemplo, aunque dada la

modularidad que se pretende del sistema debe ser intercambiable con cualquier algoritmo o método de localización existente.

### **3.1.2.2      *Características de los usuarios***

El sistema debe tener un grado de usabilidad alto, para que cualquier tipo de usuario sea capaz de aprender a usar el sistema en cuestión de minutos.

Además de los usuarios que usan el sistema para localizarse, habrá otros usuarios que serán los encargados de calibrar y gestionar el sistema. Estos dos tipos de usuarios tendrán distintos accesos y servicios proporcionados por el sistema, mucho más restrictivos para aquellos que no sean los gestores o calibradores del mismo.

### **3.1.2.3      *Restricciones***

Se pretende que sea un sistema multiplataforma y ligero en el lado del usuario. Sin la necesidad de instalación de hardware adicional, y con el software proporcionado para la conexión al sistema.

El usuario móvil también necesitará disponer de los medios para la captura de la potencia de señal recibida de los puntos de acceso en la tarjeta inalámbrica del dispositivo móvil del usuario.

## **3.1.3 *Requisitos Específicos***

### **3.1.3.1      *Requisitos Funcionales***

Req-01 Se necesitará un plano para poder mostrar la localización de forma gráfica al usuario.

Req-02 El mapa será un fichero en formato gif o jpg.

Req-03 No es necesario que se indique en el mapa ningún tipo de mobiliario.

Req-04 El mapa será en 2D, de una única altura o planta.

Req-05 Deberá haber un número suficiente de puntos de acceso cubriendo la planta.

Req-07 El mapa será dividido en zonas, que serán marcadas por sus correspondientes coordenadas.

Req-08 El gestor/calibrador determinará en cuántas zonas debe dividir el mapa, almacenando esta información en el sistema.

Req-09 Las coordenadas delimitarán un rectángulo, mediante el uso de coordenadas 2D, donde el eje de coordenadas comenzará en la esquina superior izquierda. Coordenadas de definición del rectángulo  $[(x_a, y_a)(x_b, y_b)]$ .

Req-10 El calibrador/gestor podrá configurar la localización, proporcionando el mapa de la misma y las zonas.

Req-11 Las coordenadas de las zonas deben venir en formato de enteros.

Req-12 Las potencias de las señales se almacenan en dBms.

Req-13 Una vez el usuario este conectado al sistema de localización, podrá solicitar su posición al sistema en cualquier momento.

Req-14 El usuario solo podrá estar en una de las zonas definidas por el calibrador.

Req-15 La localización del usuario será realizada por el sistema mediante los datos de potencia de señal recibida por el usuario de los distintos puntos de acceso captados.

Req-16 La información de la potencia de señal recibida por el usuario deberá mandarse junto con su petición de localización.

### **3.1.3.2 *Requisitos Interfaces externos***

Las interfaces necesarias son las de comunicación, conexión a la red. Es deseable que la comunicación se realice vía *WiFi*, aunque también pueden realizarse a través de una red de área local cableada. Dicha comunicación con el sistema debe ser transparente a la aplicación.

También será necesario en el caso de usuarios móviles o calibradores el poseer una interfaz *WiFi* para poder obtener los datos necesarios para la localización.

### **3.1.3.3 *Requisitos Rendimiento***

No tiene un límite específico en el número de usuarios, independientemente del tipo que sean. El sistema deberá dar acceso a cada usuario lo antes posible, sin que estos se queden esperando porque haya otros usuarios.



### **3.1.3.4    *Requisitos Tecnológicos***

El usuario móvil y el calibrador, deberán tener un dispositivo móvil con Tarjeta *WiFi* 802.11b ó g.

El sistema deberá instalarse en un ordenador con:

- Procesador: Pentium IV 1.6 GHz
- Memoria: 512MB
- Espacio en disco de 1GB.

### **3.1.3.5    *Seguridad***

Cuando cualquier usuario intente conectarse al sistema de localización deberá autenticarse, con su identificación (*login*) y su clave de acceso (*password*), y el sistema deberá comprobar que se trata de un usuario autorizado. Si el identificador introducido no corresponde a un usuario o la clave no coincide con la almacenada, no se le permitirá la conexión.

## **3.1.4 *Análisis de alternativas***

### **3.1.4.1    *Descripción General.***

Teniendo en cuenta el ámbito y funcionalidad que debe cubrir el sistema, se tendrán que analizar los posibles subsistemas que son necesarios, así como los requisitos a los que dan respuesta, su cobertura geográfica, de procesos y datos, así como su gestión. De manera tal que se pueda valorar una serie de alternativas de implementación del sistema y poder elegir la más adecuada para este caso.

De la sección anterior se puede concluir que el sistema va a necesitar realizar las siguientes tareas:

- Comunicación/Gestión con/de los usuarios.
- Almacenamiento de los datos necesarios para la localización.
- Cálculo de la localización de los usuarios.

Por otro lado los usuarios, sean de la clase que sean, del sistema, necesitaran:

- Tener acceso a los datos obtenidos por la tarjeta de interfaz inalámbrica de red.
- Presentación de la Localización.
- Comunicación con el Servidor.

Este resumen de tareas, da una visión global de cómo podría visualizarse el Sistema de localización que se pretende implementar.

### **3.1.4.2 Alternativas**

En este apartado se pretende decidir la mejor opción de diseño para el sistema partiendo de las funcionalidades y restricciones que posee tal y como se ha visto en los apartados anteriores de este capítulo.

Se ha visto que el Sistema debe ser modular por lo que se evoca una arquitectura basada en el Modelo de n-Capas. El cual resultaría bastante eficiente dada sus características:

- Sistema distribuido.
- Multiplataforma.
- Altamente modulable.
- Aplicaciones robustas.
- Mantenimiento y soporte sencillo.
- Flexibilidad.
- Alta escalabilidad.

Por otro lado, las características del sistema hacen hablar de un programa que proporciona servicios a través de un protocolo definido. Lo que recuerda a un sistema monolítico Cliente/Servidor.

Si tenemos en cuenta estas dos alternativas y la evolución del clásico modelo Cliente/Servidor a una arquitectura cliente/servidor de 3 o N capas, donde el servidor divide el servicio en 3 o más capas, generalmente una capa de presentación, otra de lógica de negocio, y la tercera y última de acceso a datos. Se elige esta última arquitectura para el sistema, la también conocida como MVC-3tier, Modelo Vista Controlador de 3 capas.

Para ganar en eficiencia se debería decir que el cliente a diseñar debería ser un cliente ligero, en el cual solo tuviese que tener la presentación, haciendo uso de todos los recursos en el sistema, al contrario de los clientes pesados. Pero la restricción de acceso a los drivers de las tarjetas inalámbricas, obligan a invocar métodos de lectura de la señal de APs desde el cliente; por lo que se convierte en un cliente híbrido (hace uso del menor uso de recursos posibles, pero tiene más capas que la de presentación).

## 4 Diseño

### 4.1 Objetivo del Sistema

Como se ha visto en el Capítulo 3, el sistema se va a desarrollar siguiendo la arquitectura cliente-servidor, basada en el modelo MVC (Modelo Vista Controlador). Servirá para recolectar datos de calibración en un entorno inalámbrico y realizar el procesado mediante métodos de Posicionamiento para proveer al usuario de una posición dentro de una localización.

#### 4.1.1 Contexto

El motor de localización o sistema, se denominará, *WifiTool*. Éste interactuara con tres tipos de actores, “*Gauge*”, “*User/Mobile*” y “*Monitor*”, como se muestra en la Figura 9.

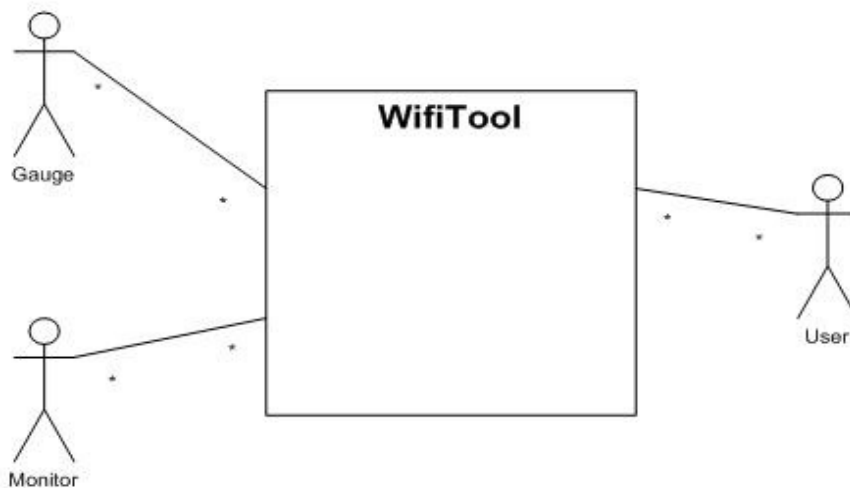


Fig. 9: Diagrama de contexto

- Gauge (calibrador): es el encargado de calibrar/configurar el sistema para poner en funcionamiento el sistema en una nueva localización.

Su tarea principal será obtener el mapa y la calibración por zonas de la nueva localización e inicializar un nuevo registro en el servidor para hacerle saber:

- Nombre de la nueva localización y el mapa de la misma, así como la división en zonas del mapa de dicha localización.
- Los datos de la calibración de las diferentes zonas en las que se ha dividido la localización.

También será capaz de realizar las siguientes consultas/funciones:

- Saber las localizaciones existentes en el sistema.
- Podrá borrar las localizaciones junto con su correspondiente calibración.
- User: es el usuario del sistema (móvil), solicitará una localización de su posición.

Se considerara User a cualquier equipo con conexión inalámbrica 802.11 que se encuentre en dicha localización. Sus contactos con el servidor serán los siguientes.

- Descubrimiento, localización a la que pertenece.
- Datos de situación (AP's – potencias).

El usuario puede también funcionar activando la posibilidad de que sea un usuario localizable. En ese caso mandará al sistema sus datos, por defecto cada 30 segundos, para poder ser localizado dentro de la red, si en 30 minutos (como valor por defecto) no ha actualizado su localización el sistema borrará sus datos considerando que ya no se encuentra disponible en la red. Ambos valores podrán ser configurables en el sistema.

- Monitor: es el usuario de monitorización del sistema (fijo/móvil), capaz de visualizar la posición de todos los usuarios móviles registrados en el sistema. El único que no necesita conexión inalámbrica.

El sistema le envía la información necesaria sobre la localización donde quiera ver los usuarios. Obteniendo la siguiente información:

- Mapa de la localización.
- Coordenadas de las zonas en las que se divide el mapa, con el siguiente formato para cada zona:

<X1;Y1;X2;Y2>

Coordenadas que definen un rectángulo dentro de un sistema de coordenadas donde (0,0) corresponde a la esquina superior izquierda.

- Dirección MAC de los usuarios móviles registrados en el sistema en ese momento.
- Zona donde se encuentra cada uno de los usuarios.

### 4.1.2 Casos de Uso

El diagrama de casos de uso de Wifitool es el que se muestra en la figura 10.

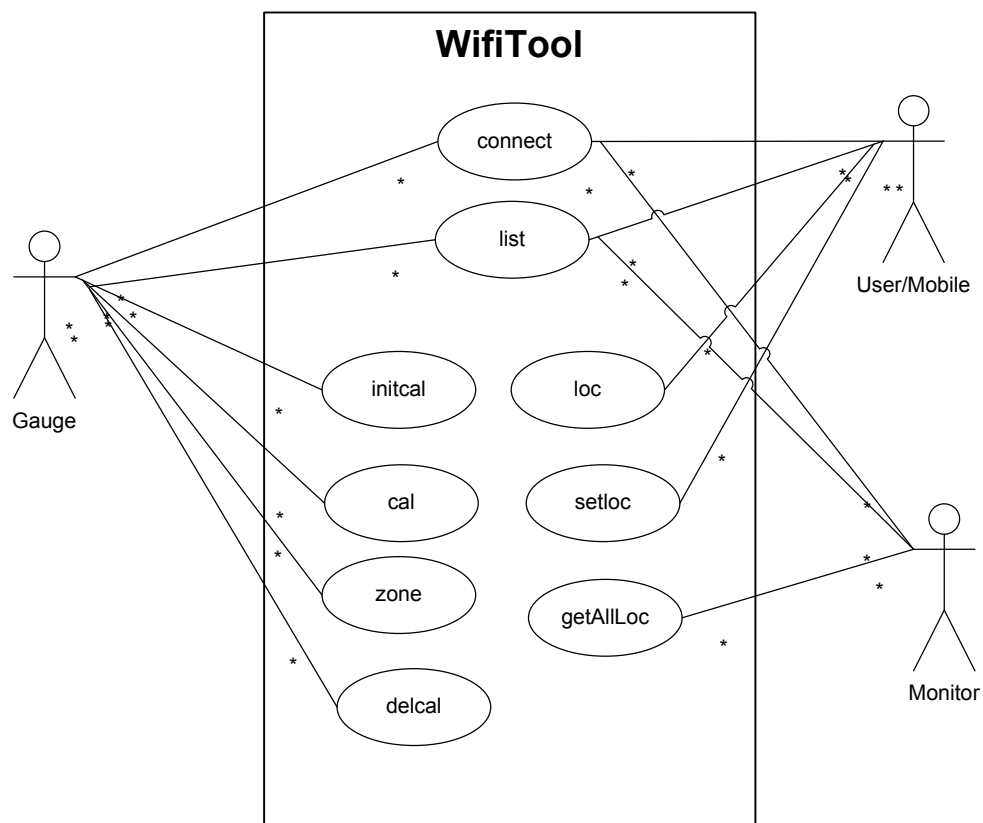


Fig 10: Casos de Uso.

i) *Caso de uso: Connect.*

Petición al sistema de la conexión a sus servicios y datos. Se le requerirá autenticación en sistema (user/passwd) a todos los usuarios.

ii) *Caso de uso: Initcal.*

Gauge inicializa la calibración de sistema para una nueva localización. Para ello se necesita configurar el sistema, por lo que se creará una entrada nueva básica para la localización, Gauge proporcionará al sistema un nombre para la localización y su mapa, con las correspondientes coordenadas a las zonas en las que se divide éste, para posteriormente poder calibrar dicha localización.

iii) *Caso de uso: Cal.*

Gauge calibra una zona del sistema mediante 'cal', envía los datos al servidor de la calibración que se esté realizando a una localización. Se mandan al sistema los siguientes datos:

- Dirección MAC de los APs detectados.
- Potencia en dBm de cada APs.
- Zona en la cual se están capturando los datos.

iv) *Caso de uso: Delcal.*

Gauge borra del sistema toda la calibración existente para una localización, así como su configuración. Se hará uso de este caso cuando se quiera eliminar una localización del sistema, o bien se quiera reconfigurar las zonas en las que se divide dicha localización o cambiar de mapa o nombre. Para ello se hará una eliminación completa de dicha localización del sistema, siendo necesario volver a configurarla, mediante 'initcal' y calibrarla mediante 'cal'.

v) *Caso de uso: Zone.*

Gauge recupera el número de zonas en las que se encuentra dividido el mapa de una localización en concreto.

vi) *Caso de uso: List.*

Cualquier usuario puede obtener el listado de localizaciones en el sistema.

vii) *Caso de uso: Loc.*

User/Mobile solicita su localización proporcionando al sistema los datos obtenidos mediante su interfaz inalámbrica.

viii) *Caso de uso: setloc.*

User/Mobile actualiza los datos de su dispositivo inalámbrico para poder ser localizable.

ix) *Caso de uso: getLocAll.*

Monitor obtiene los datos de localización de todos los usuarios registrados en el sistema en ese momento para una localización en concreto. Cada usuario será identificado mediante su dirección MAC, ya que es un identificador único.

## ***4.2 Diseño Lógico***

### ***4.2.1 Estructura Estática del sistema***

#### ***4.2.1.1 Sistema***

El servidor/sistema será el encargado de almacenar los datos de las distintas localizaciones y situar a los distintos usuarios en la red.

Para cumplir con el objetivo de modularidad el servidor se divide en tres subsistemas:

- **SERVER**

Subsistema encargado de controlar el protocolo de comunicaciones con los usuarios, así como de controlar el funcionamiento del sistema.

- **DB ACCESS**

Es el subsistema encargado del tratamiento de los datos, su almacenamiento y manejo de forma persistente en una base de datos.

- **CLASSIFIER**

Este subsistema se encargará de posicionar los usuarios del sistema mediante distintos algoritmos de posicionamiento.

En la siguiente figura se puede apreciar como quedaría la Arquitectura del sistema.

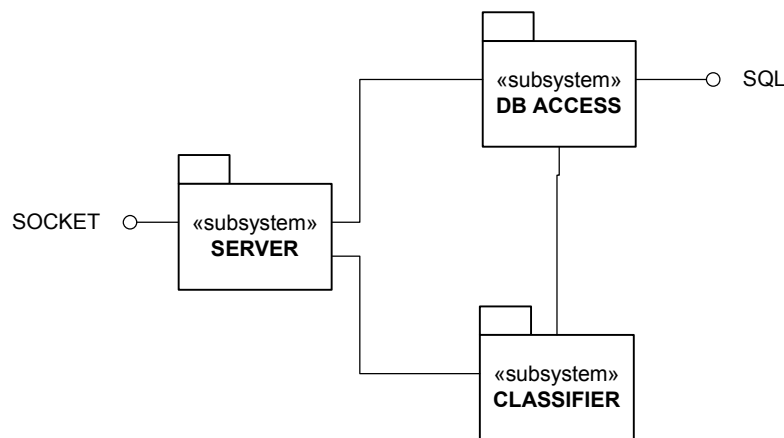


Fig 11: Arquitectura del Sistema.

#### 4.2.1.2 Componentes

Tabla 1: Componente “SERVER”

Componente:	SERVER
Responsabilidades:	A través de este componente se accederá al Sistema. Proporciona el protocolo de comunicación del sistema. Se podría decir que es el componente “Controlador” de un modelo MVC (Modelo Vista Controlador). Obtiene los datos de los usuarios móviles. Calibra el sistema. Proporciona las localizaciones de los usuarios.
Colaboradores	<ul style="list-style-type: none"> <li>• DB ACCESS</li> <li>• CLASSIFIER</li> </ul>
Notas:	Puede ser accedido por múltiples usuarios simultáneamente. Funcionará con un hilo por usuario para tratarlos concurrentemente.
Incidencias:	

Tabla 2: Componente “DB ACCESS”

Componente:	DB ACCESS
Responsabilidades:	Proporciona al sistema los datos mediante el acceso a una base de datos centralizada que poseerá el sistema.
Colaboradores	<ul style="list-style-type: none"> <li>• SERVER</li> <li>• CLASSIFIER</li> </ul>
Notas:	Se crea 1 instancia de DB ACCESS por cada hilo que crea el SERVER. SERVER será el encargado de controlar dicha instancia, de su creación y destrucción.
Incidencias:	



Tabla 3: Componente “CLASSIFIER”

Componente:	CLASSIFIER
Responsabilidades:	Proporciona la situación de los usuarios, aplicando un algoritmo de localización que obtiene de los datos de calibración que serán accedidos mediante DB ACCESS, y de los datos de usuario.
Colaboradores	<ul style="list-style-type: none"><li>• SERVER</li><li>• DB ACCESS</li></ul>
Notas:	Cada instancia de CLASSIFIER corresponde a una instancia de SERVER. Hará uso a la instancia de DB ACCESS correspondiente al SERVER para acceder a los datos de calibración.
Incidencias:	

### 4.2.1.3 Interfaces

Tabla 4: Interfaz “SOCKET”

Interfaz:	SOCKET
Descripción:	Comunicación con los actores del sistema.
Servicios:	Entrada/salida de datos al sistema.
Protocolos:	Sockets sobre TCP.
Notas:	Se montara sobre los sockets un Protocolo para comunicarse los usuarios con el sistema. Se accede a este interfaz mediante el componente “SERVER”.

Tabla 5: Interfaz SQL.

Interfaz:	SQL
Descripción:	Almacenamiento de datos persistentes.
Servicios:	Almacena y recupera los datos de los usuarios y de las calibraciones.
Protocolos:	SQL
Notas:	Acceso mediante DB ACCESS, necesario instalar un SQL SERVER.

## 4.2.2 Estructura Estática del cliente

### 4.2.2.1 Cliente

Los posibles clientes del sistema deberán realizar la función de “Vista” de un modelo vista controlador (MVC). Por lo que deberán diseñarse con dos componentes básicos para su funcionamiento:

- Presentation.

Será el componente encargado de la visualización del sistema. Es decir, se encargará de la Presentación del sistema y la comunicación con el mismo.

- WiFi-Access:

Este módulo deberá interconectar con la tarjeta inalámbrica del usuario para obtener los datos de la red en la posición deseada. Conectará con dicha tarjeta a través de un módulo externo que llamaremos ‘wifipos’.

En la siguiente figura se puede apreciar como quedaría la Arquitectura del sistema:

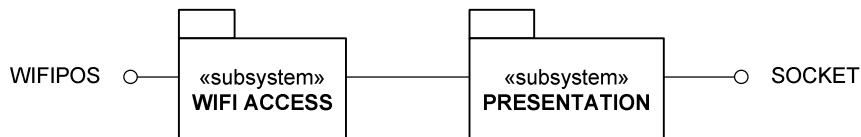


Fig 12.- Arquitectura del Cliente.

### 4.2.2.2 Componentes

Tabla 6: Componente “PRESENTATION”

Componente:	PRESENTATION
Responsabilidades:	<p>A través de este componente se realizará la visualización y comunicación con el sistema; a través del protocolo propietario de comunicación del mismo. Se podría decir que es el componente “Vista” de un modelo MVC (Modelo Vista Controlador).</p> <p>Su funcionalidad dependerá del actor que este haciendo uso del sistema.</p> <p>Podrá tener acceso a la tarjeta WiFi a través del componente WiFi Acces.</p>
Colaboradores	<ul style="list-style-type: none"> <li>• WIFI ACCESS (opcional)</li> </ul>
Notas:	Para el usuario “Monitor” no será necesario el uso del componente “WIFI ACCESS”
Incidencias:	

Tabla 7: Componente “WIFI ACCESS”

Componente:	WIFI ACCESS
Responsabilidades:	Este componente es el encargado de obtener los datos de la red inalámbrica. De los datos obtenidos mediante la módulo externo ‘wifipos’: MAC- Conexión- Nombre-Potencia; solo se queda con el identificador único MAC y la Potencia de la señal.
Colaboradores	<ul style="list-style-type: none"><li>• PRESENTATION</li></ul>
Notas:	En el caso de nuestro cliente pruebas, se comunicará con el programa ‘wifipos’ [2.1.3.6], el cual será el que acceda directamente a la Base de Datos.
Incidencias:	

### 4.2.2.3 Interfaces

Tabla 8: Interfaz SOCKET.

Interfaz:	SOCKET
Descripción:	Comunicación con el sistema.
Servicios:	Entrada salida de datos al cliente.
Protocolos:	Sockets sobre TCP.
Notas:	Se montara sobre los sockets un Protocolo para comunicarse los con el sistema. Se accede a este interfaz mediante el componente “Presentation”.

Tabla 9: Interfaz con ‘Wifipos’

Interfaz:	Con Wifipos
Descripción:	Volcado por salida/entrada estándar de la información de la red inalámbrica obtenida a través de la tarjeta inalámbrica. Obteniendo los siguientes datos: MAC Conexión Name Potencia
Servicios:	Comunicación con la tarjeta inalámbrica.
Protocolos:	I/O estándar
Notas:	Comunicación a través de ‘Wifipos’ con la tarjeta inalámbrica.

#### Nota:

Debido a la dificultad de acceso a la tarjeta inalámbrica, distintos drivers e interfaces de conexión en función del S.O., se accederá a ella a través de un módulo externo que llamaremos ‘Wifipos’. Este módulo será el encargado de realizar las llamadas al sistema y se comunicara con el cliente a través del módulo WiFi-Access que será capaz de hablar con cualquiera de los ‘Wifipos’. El protocolo usado para hablar con la tarjeta inalámbrica es NDIS.

## ***Wifipos***

*Wifipos* es un programa escrito en lenguaje de bajo nivel que cumple la única función que no puede realizarse desde *Java* – acceder a las funcionalidades ofrecidas por los controladores de las tarjetas inalámbricas, con el objeto de recuperar las distintas potencias recibidas en la tarjeta con respecto a distintas fuentes de señal.

Para acceder a los datos de la tarjeta se deben realizar acceso a funcionalidades de bajo nivel a través de una *API*, que desgraciadamente, dependerá del Sistema Operativo en el que este instalada la tarjeta.

Para el caso de cliente corriendo bajo S.O. Microsoft Windows, *Wifipos* será un ejecutable ‘wifipos.exe’, desarrollado en Visual C++ con llamadas a la *API* de *NDIS* (Network Driver Interface Specification).

En concreto, dentro de *NDIS* se utiliza la función GET para solicitar al sistema el parámetro cuyo OID es 802\_11\_RSSI que devuelve, como su nombre indica, el indicador de potencia de señal recibida *RSSI* (Received Signal Strength Indication). Estos indicadores, que se proporcionan por punto de acceso, permiten comparar las potencias de señal recibidas para cada red a la que el dispositivo inalámbrico tiene acceso. Ha de tenerse en cuenta que la potencia se mide en unidades relativas – el estándar 802.11 no especifica cuál debe ser esa unidad (siendo lo más normal el dBm o el mW) y sólo garantiza que el valor debe ser representable en un entero de 8 bits.

Una vez se ha obtenido la información para todas las tarjetas inalámbricas de un terminal y para todas las redes disponibles en el entorno, esta información se redirige a la salida estándar, desde donde puede ser fácilmente interceptada por cualquier programa que lo desee.

Para clientes corriendo bajo Linux/Unix, *Wifipos* será un programa en perl ‘wifipos.pl’, que realiza llamadas al sistema para acceder a la tarjeta. Dichas llamadas al sistema acceden a los datos a través de la *API NdisWrapper*. El funcionamiento y la salida de del programa serán idénticos a los obtenidos por ‘Wifipos.exe’.

## ***4.2.3 Comportamiento dinámico***

### ***4.2.3.1 Escenarios – Protocolo de comunicación***

#### ***i) Connect***

En la figura (13), se puede ver un escenario de éxito para el caso de uso “Connect”. Primero se realiza una conexión entre cliente servidor, mediante sockets. Una vez que dicha conexión se ha realizado se le comunica al cliente mediante un OK y se pasa a solicitar la validación en el sistema, petición de user/passwd.

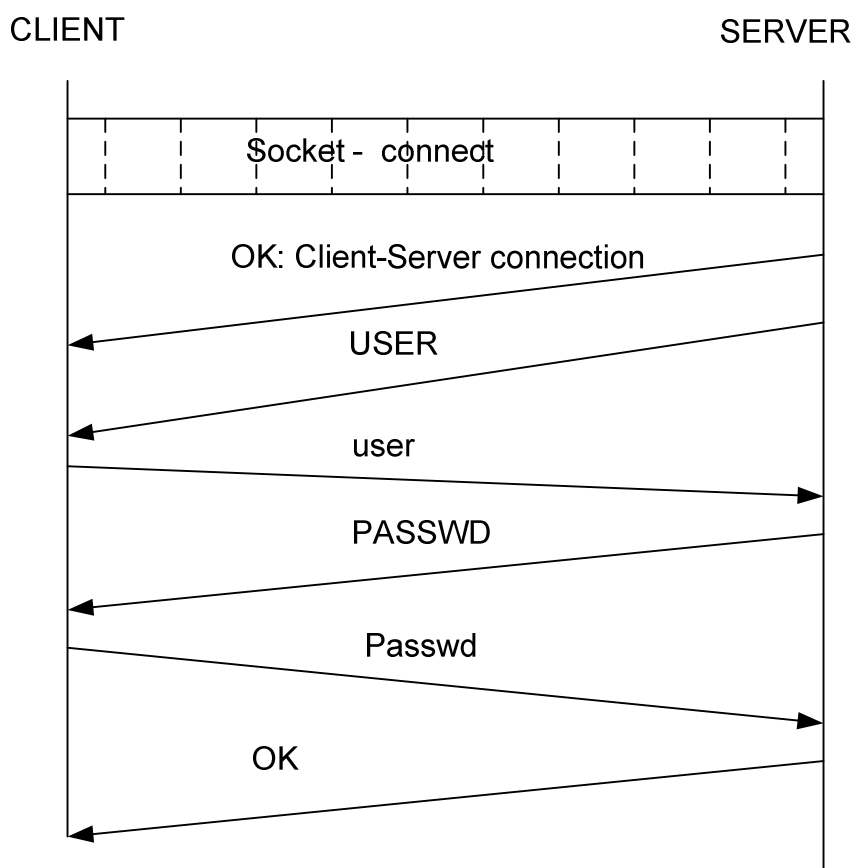


Fig 13: Establecimiento de Conexión realizado con éxito.

En la siguiente figura (14), se ve un caso de fallo en “Connect”. Una vez realizada la conexión mediante sockets, ocurre un fallo en la conexión de la base datos, por lo que no se puede continuar dicha conexión.

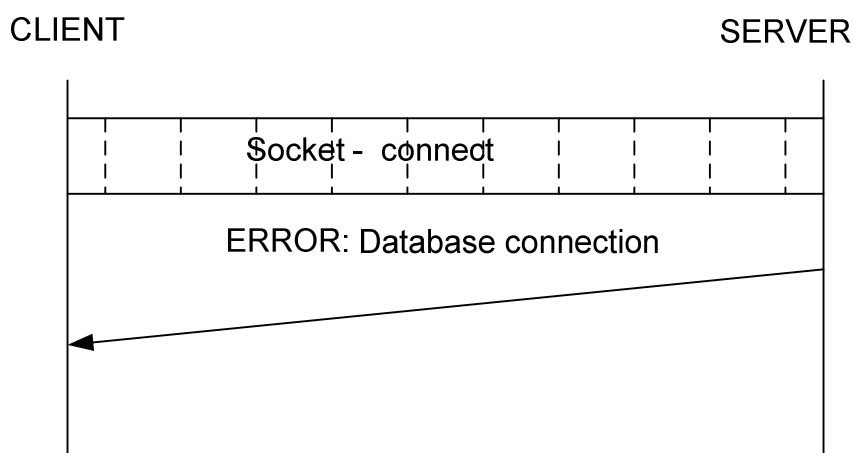


Fig 14: Establecimiento de conexión con errores.

En la figura (15), se ve otro caso de fallo, en el cual, la identificación del cliente no es la adecuada, error en el usuario o la contraseña.

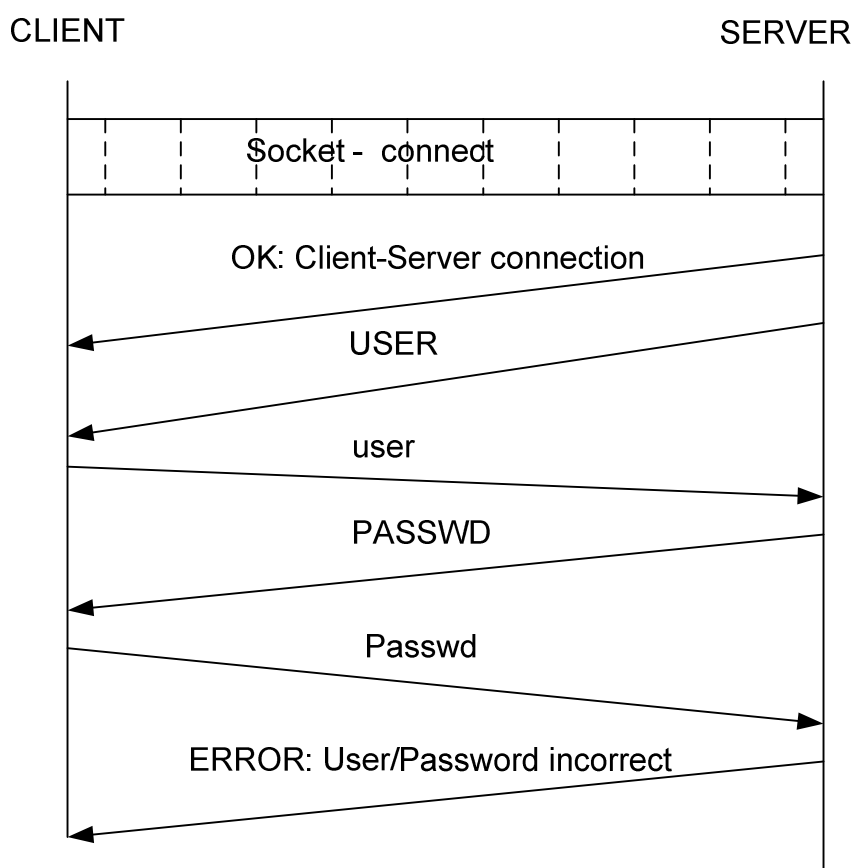


Fig 15: Establecimiento de conexión con errores.

## ii) *Initcal: Configuración/inicialización de la calibración*

Con el caso de uso initcal podemos configurar una localización por lo que mandaremos el siguiente comando al servidor:

*Initcal* <Location> <map> <xa1;ya1;xb1;yb1;...;xan;yan;xbn;ybn>

- Location: Nombre de la nueva localización que se va a calibrar.
- Map: Fichero que contiene el plano de la Localización a calibrar, en formato .gif o .jpg.
- <xa1;ya1;...xan;yan;xbn;ybn>: Coordenadas en las que se divide el mapa de la localización en zonas (Ej: zona A <xa;ya,xb,yb>). Las zonas serán rectángulos que se definen con dichas coordenadas sobre el mapa, siendo el inicio de coordenadas la esquina superior izquierda.

El servidor almacena dichos datos y procede a solicitar el fichero del mapa para almacenarlo en el sistema, mediante el comando

*REQ-MAP* <loc>

, como se puede apreciar en la siguiente figura (16).

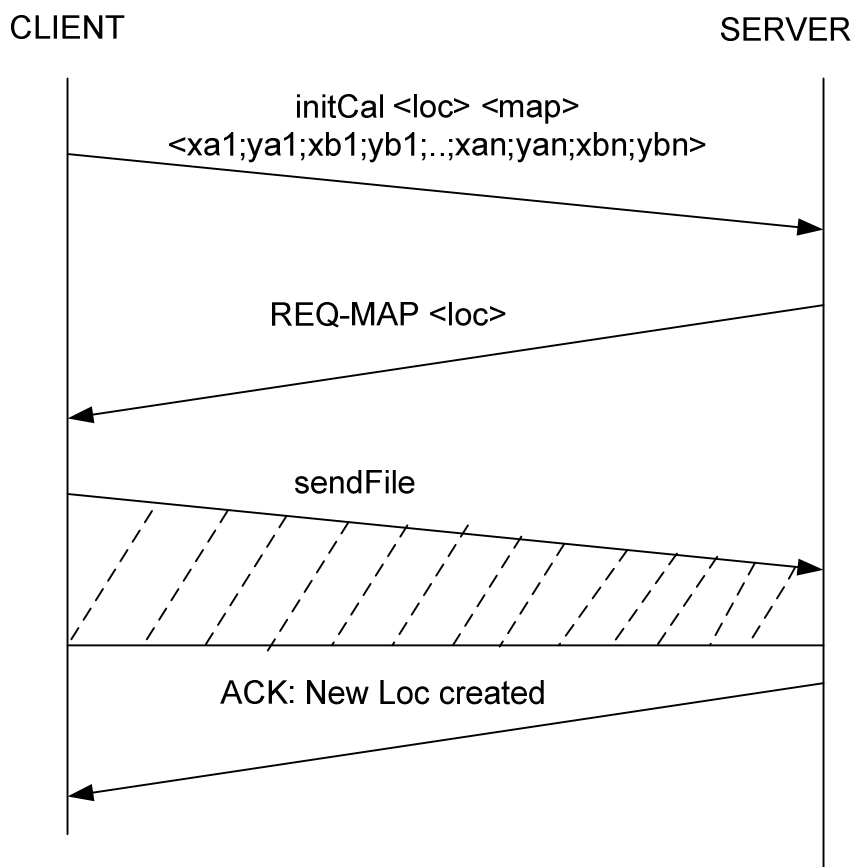


Fig 16: Initcal (Configuración) realizada con éxito.

En la siguiente figura (17) se ve un caso de fallo de “initcal”, en el que dicha localización ya existe, por lo que no se puede inicializar una nueva; y el sistema devolverá el comando ‘ERROR’ con la explicación del fallo.

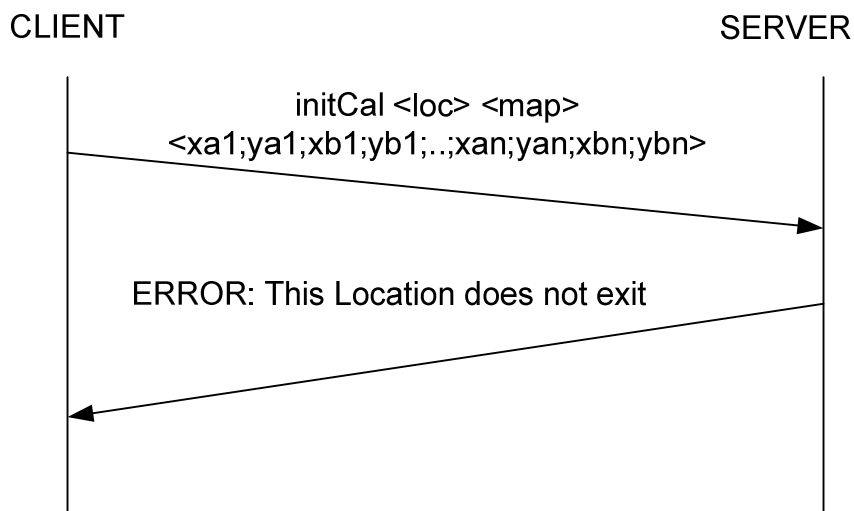


Fig 17: Initcal (Configuración) realizada con errores.

En el siguiente escenario de error, el servidor indica que no ha podido iniciar la calibración debido a un fallo en los datos o la base de datos, figura (18).

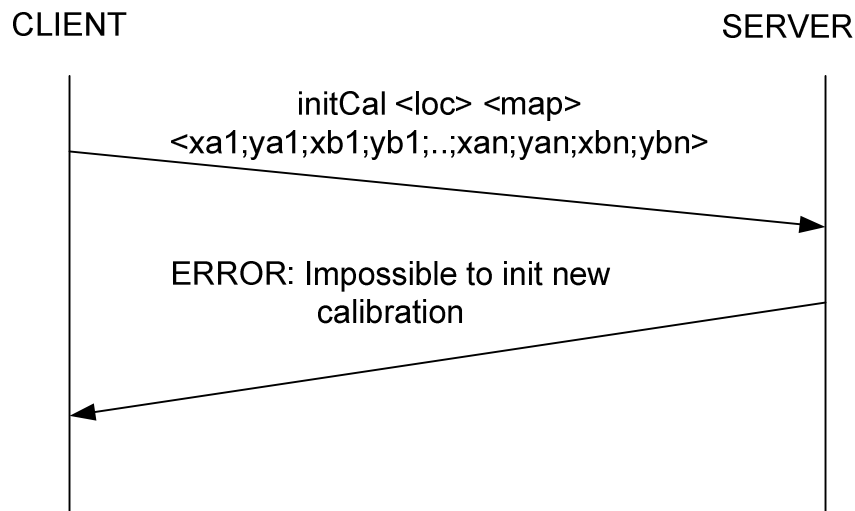


Fig 18: Initcal (Configuración) realizada con errores.

### iii) *Cal: Calibración*

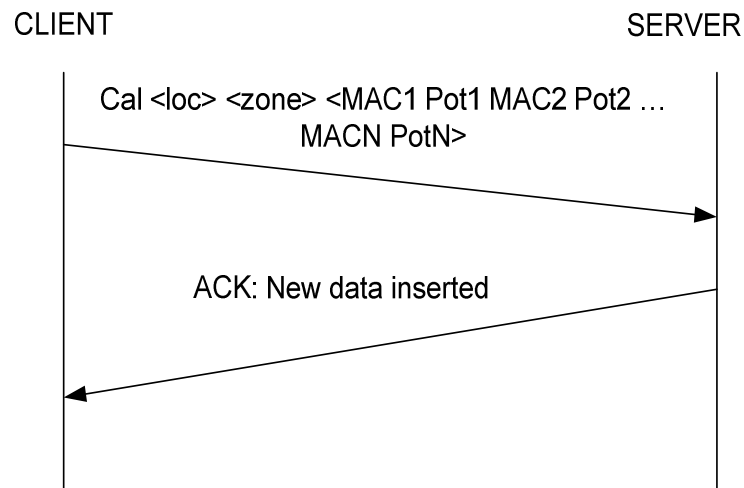
Con el caso de uso “cal”, se calibrará la localización, mandando los datos adquiridos del entorno en las distintas zonas al sistema:

*Cal*     <Location> <zone> *MAC1 Pot1 ... MACN PotN*

- Location: Nombre de la Localización que se está calibrando
- Zone: Zona que se está calibrando.
- *MACx Potx* : Colección medidas de Direcciones y Potencias tomadas en zona, separadas por espacio. *MACx* será la dirección MAC de las APs y *PotN* será la potencia en dBm recibida por dicho AP.

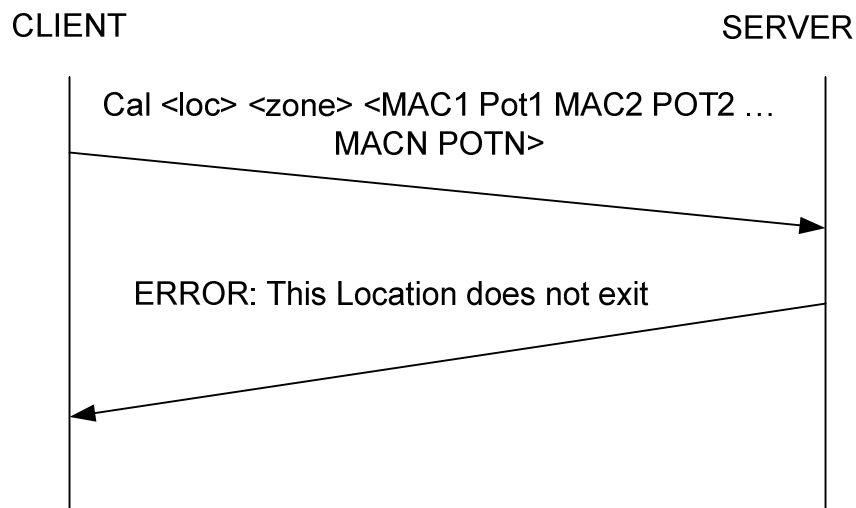


En la siguiente figura (19) se puede apreciar un caso de éxito de una calibración, el sistema devuelve el comando ACK para indicarlo.



**Fig 19: Cal (Calibración) realizada con éxito.**

En la siguiente figura (20) se puede apreciar un caso de error de “cal”. Debido a que la Localización que se intenta calibrar no está en el sistema, por lo que es necesario iniciar la calibración de dicha localización.



**Fig 20: Cal (Calibración) realizada con errores.**

En la siguiente figura (19) se puede apreciar un escenario de fallo de calibración porque el comando mandado no es el correcto.

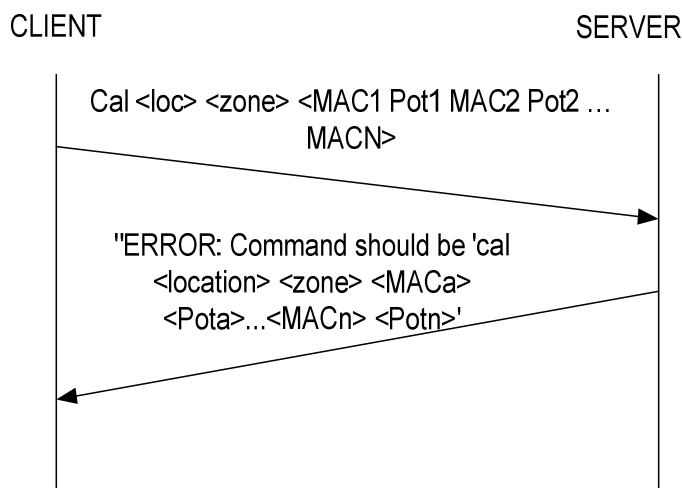


Fig 21: Cal (Calibración) realizada con errores.

En el siguiente escenario de error, el servidor indica que no ha podido calibrar la zona debido a un fallo en la base de datos, figura (22).

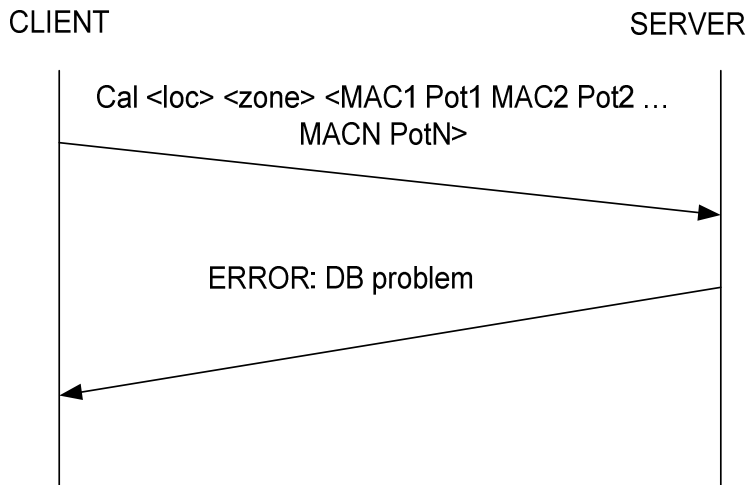


Fig 22: Cal (Calibración) realizada con errores.

#### iv) *delCal: Borrado de Calibración*

Con “delcal”, se borrarán todos los registros de una localización en el sistema:

*delCal* <Location>

En la siguiente figura (23) se puede apreciar un escenario de éxito del borrado de la localización.

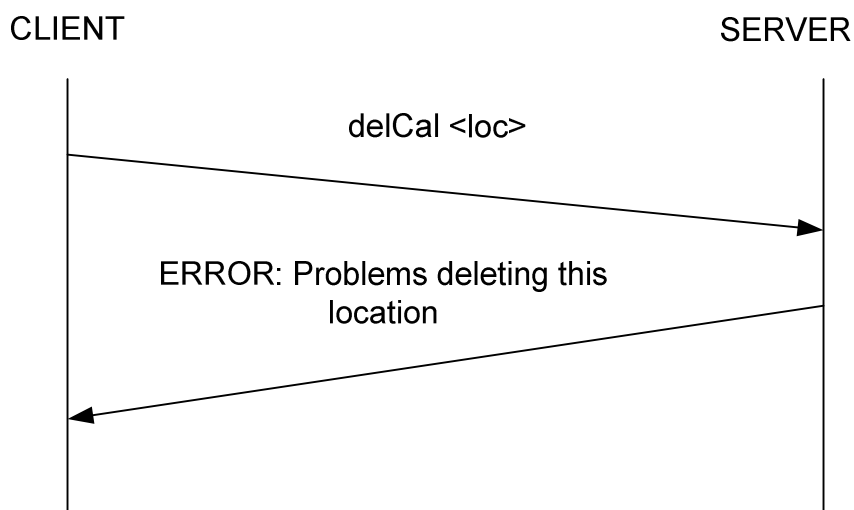


Fig 23: delCal (Borrado de Calibración) realizada con éxito.

En la siguiente figura (24) se puede apreciar un escenario de error del borrado de la localización. Puede ser debido a tres motivos:

- Problemas en la base de datos.
- Problemas al borrar las calibraciones.
- Problemas al borrar los datos de los usuarios que se encuentran en dicha localización.

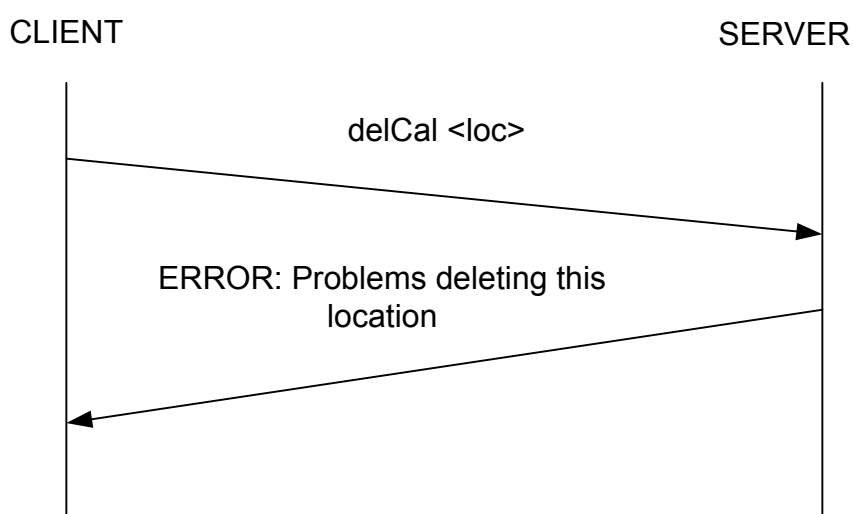


fig 24: delCal (Borrado de Calibración) realizada con errores.

v) **zone: Obtención del número de zonas de una localización.**

Con “zone”, se obtiene el número de zonas en las que se divide la localización:

*zone*     <Location>

En la siguiente figura (25) se puede apreciar un escenario de éxito de “zone”.

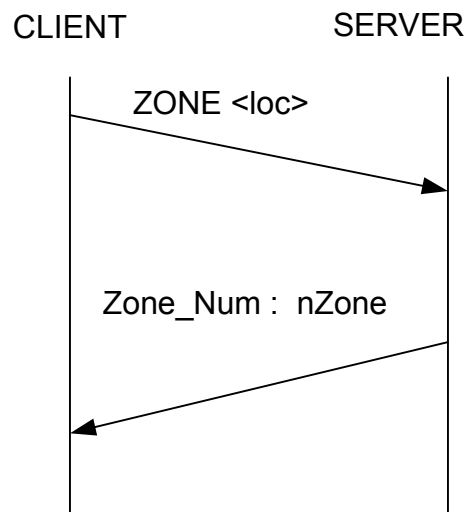


Fig 25: Zone (Obtención de número de zonas) realizada con éxito.

En la siguiente figura (26) se puede apreciar un escenario de error de “zone”.

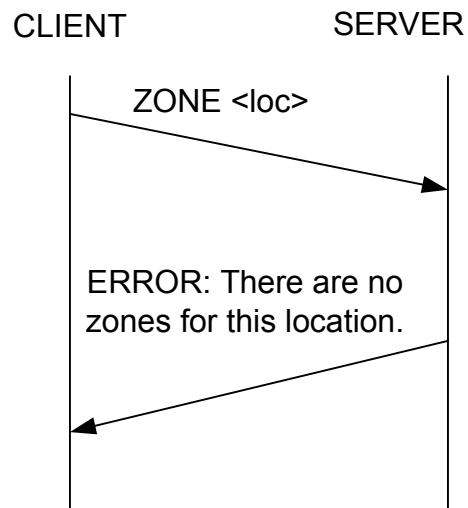


Fig 26 Zone (Obtención de número de zonas) realizada con errores.

vi) **List: Lista**

Lista las localizaciones que tiene disponibles el servidor. Se puede ver en la Figura(27).

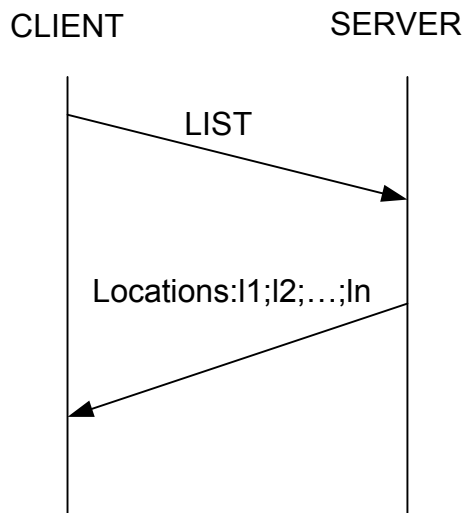


Fig 27: List realizada con éxito.

Caso de error en el caso de uso list, porque no hay localizaciones disponibles.

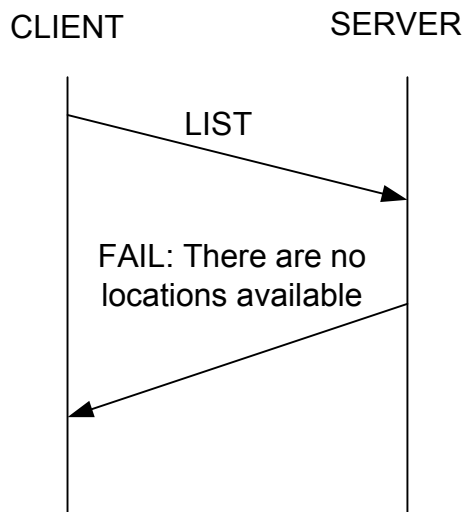


Fig 28: List con errores

**vii) Loc: Localización**

Localiza al usuario en una zona del mapa. El usuario envía los datos de su MAC y las medidas de potencia que tiene en su posición.

*loc*    *<Location> <MAC> MAC1 Pot1 ... MACN PotN*

- Location: Nombre de la Localización en la que se encuentra el usuario.
- MAC: Dirección identificativa del usuario.
- *MACx Potx* : Colección de medidas de Direcciones y Potencias tomadas, separadas por espacio.

El usuario manda a la plataforma sus datos, y obtendrá un mapa con las zonas en las que se encuentra dividida la localización, y la zona en la que está su posición.

Una vez ha recibido los datos del usuario móvil correctamente devuelve el siguiente comando:

*SEND*    *<nMap> <coord> <nZone>*

- nMap: Nombre del mapa de la localización, para indicar el envío de un fichero binario.
- coord: Coordenadas en las que se divide cada zona del mapa..
- nZone : Número de zonas en las que se divide el mapa.

El usuario móvil responderá con 'OK\_REQ' para indicarle al sistema que puede comenzar con el envío del fichero que contiene el mapa, 'nMap'. En ese momento comienza a enviar el sistema el mapa solicitado, una vez terminado dicho envío el usuario móvil indicará al sistema que le proporcione el número de zona en el que se encuentra mediante el comando 'OK\_ZONE'. El sistema le responderá con la zona mediante el comando 'ZONE'.

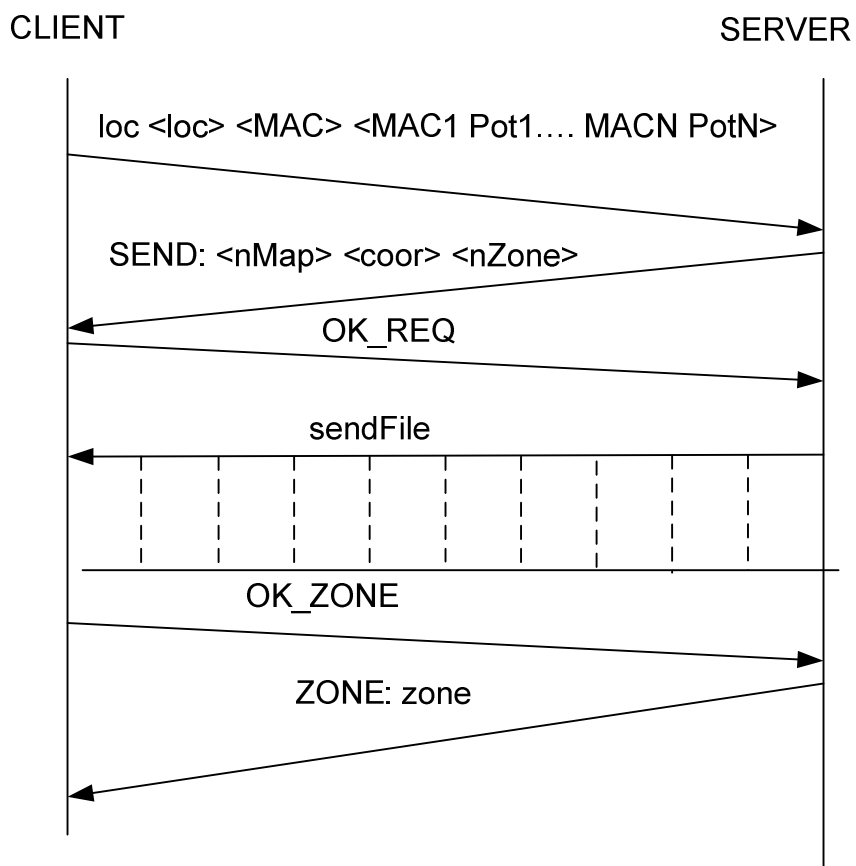


Fig 29: LOC realizada con éxito.

Escenario de error pues no existe dicha localización.

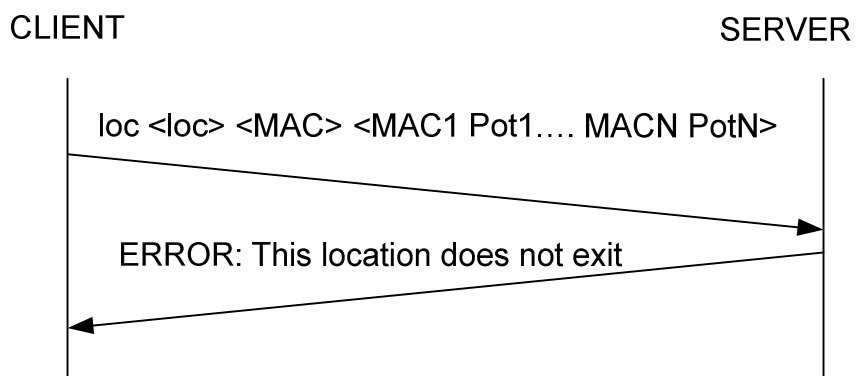


Fig 30: Loc realizada con errores.

Escenario de error en el que no se puede obtener una situación, pues dicha localización no se encuentra calibrada.

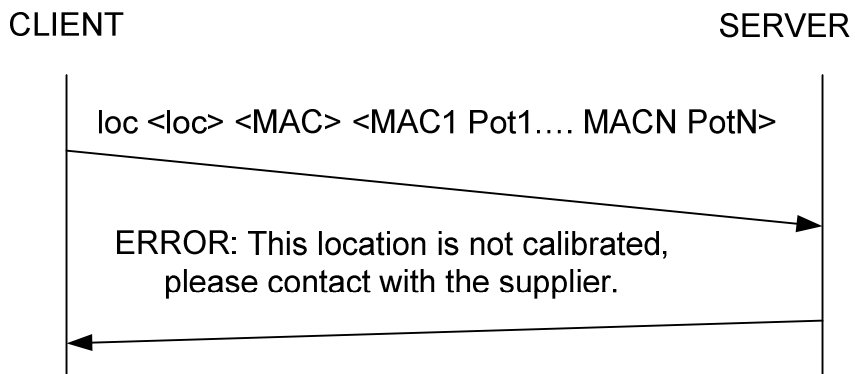


Fig 31: Loc con errores.

Escenario de error debido a un error en la base de datos.

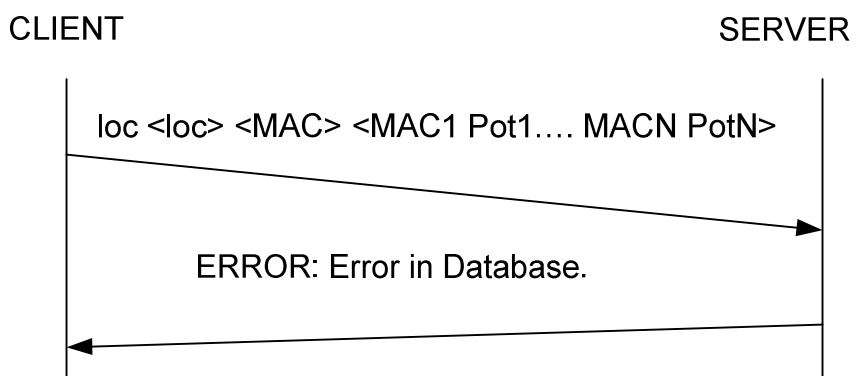


Fig 32: Loc con errores.



Escenario de fallo en el envío/recepción del mapa de la localización.

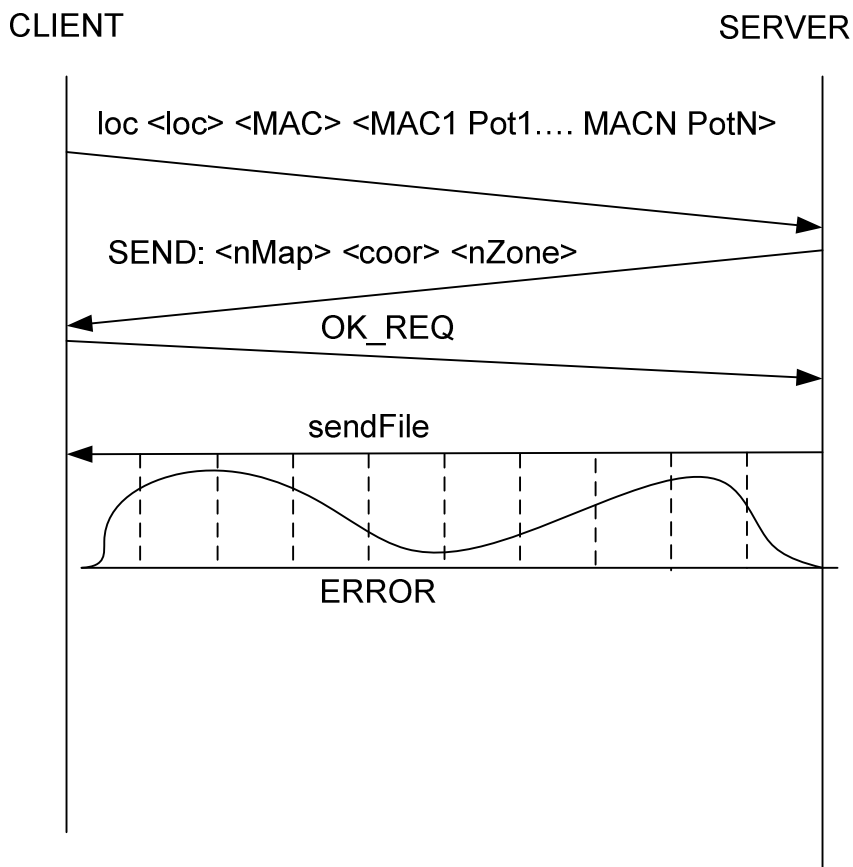


Fig 33: Loc con errores.

**viii) Setloc: Manda los datos para estar localizable.**

Setloc hace que un usuario sea localizable en dentro de una localización. El usuario envía los datos de su MAC y las medidas de potencia que tiene en su posición.

*setloc <MAC> MAC1 Pot1 ... MACN PotN*

- MAC: Dirección identificativa del usuario.
- *MACx Potx* : Colección medidas de Direcciones y Potencias tomadas, separadas por espacio.

El usuario manda a la plataforma sus datos para poder ser localizable.

En la siguiente figura (34) se puede apreciar un escenario de éxito de “setloc”.

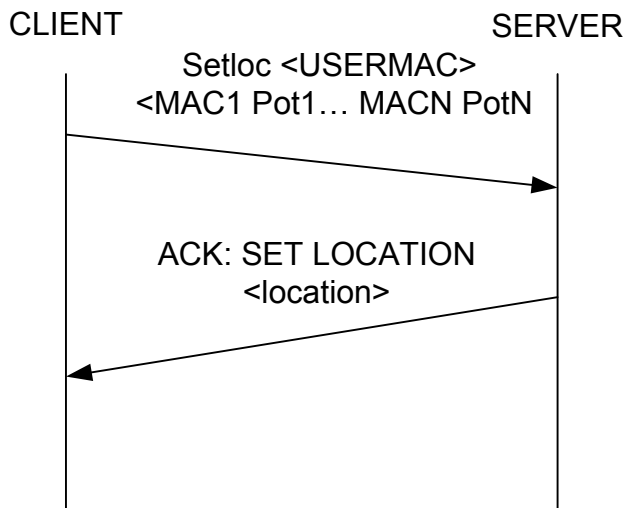


Fig 34: SETLOC realizado con éxito.

En la siguiente figura (35) se puede apreciar un escenario de error de “setloc”, debido a problemas en la base de datos.

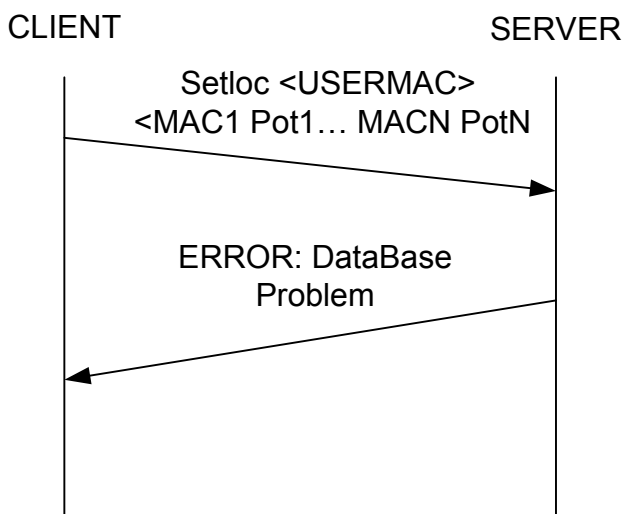


Fig 35: SetLoc realizado con errores.

En la siguiente figura (36) se puede apreciar un escenario de error de “setloc”, debido a que no existe la localización en la que se encuentra el usuario.

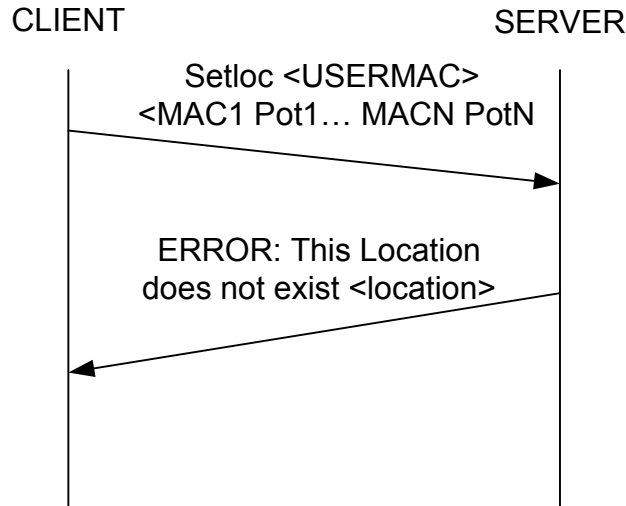


Fig 36: SetLoc con errores.

**ix) *getAllLoc: Localización de todos los usuarios.***

Localiza a los usuarios registrados en el sistema que se encuentran en una localización durante la última media hora (o el tiempo que se encuentre configurado). Se solicitará la localización de usuarios en una localización en concreto:

*getAllloc* <Location>

- Location: Nombre de la Localización que de la que se quiere recuperar información.

El usuario manda la localización donde quiere monitorizar a los usuarios, obtendrá un mapa con las zonas en las que se encuentra divida la localización, y la zonas en las que se encuentran cada uno de los usuarios, identificados por su dirección MAC.

Una vez ha recibido la localización en la que se quiere posicionar a los usuarios, el sistema le responde comunicándole los datos de dicha localización:

*SEND* <nMap> <coord> <nZone>

- nMap: Nombre del mapa de la localización, para indicar el envío de un fichero binario.
- coord: Coordenadas en las que se divide cada zona del mapa..
- nZone : Número de zonas en las que se divide el mapa.

Monitor responderá con 'OK\_REQ' para indicarle al sistema que puede comenzar con el envío del fichero que contiene el mapa, 'nMap'. En ese momento comienza a enviar el sistema el mapa solicitado, una vez terminado dicho envío Monitor indicará al sistema que le proporcione el número de zona en la que se encuentran los usuarios registrados mediante el comando 'OK\_ZONE'. El sistema le responderá con la zona mediante el comando 'ZONE'.

En la siguiente figura (37) se puede apreciar un escenario de éxito de "getAllloc".

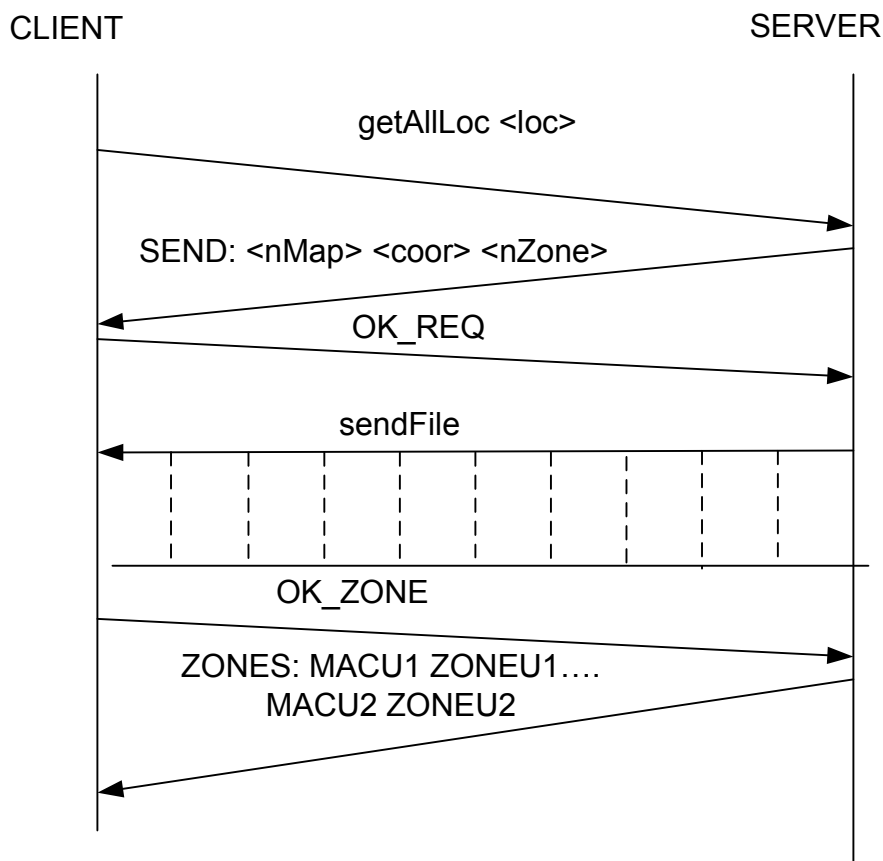


Fig 37: GetAllLoc realizado con éxito.

En la siguiente figura (38) se puede apreciar un escenario de error de “getAllLoc”, debido a que no existe la localización de la que se quiere obtener información.

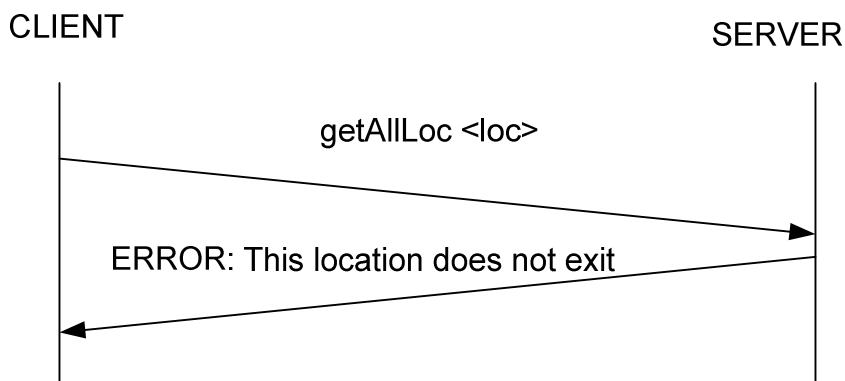


Fig 38: getAllLoc con errores.

En la siguiente figura (39) se puede apreciar un escenario de error de “getAllLoc”, debido al no haber sido previamente calibrada esa zona no es posible situar a los usuarios.

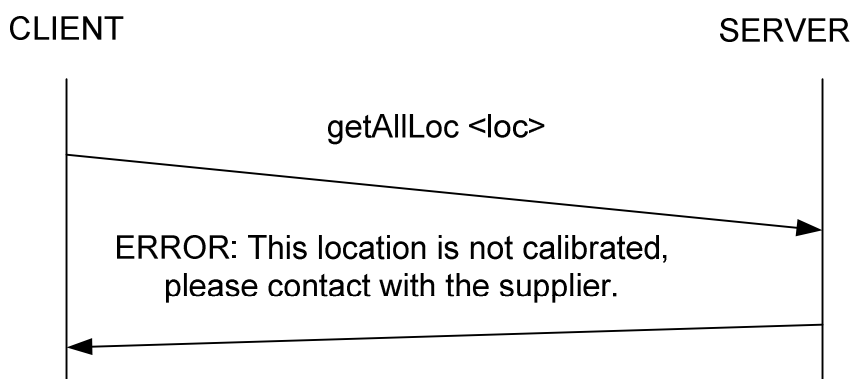


Fig 39: getAllLoc con errores.

En la figura (40) se puede apreciar un escenario de error de “getAllLoc”, debido a un problema en la base de datos.

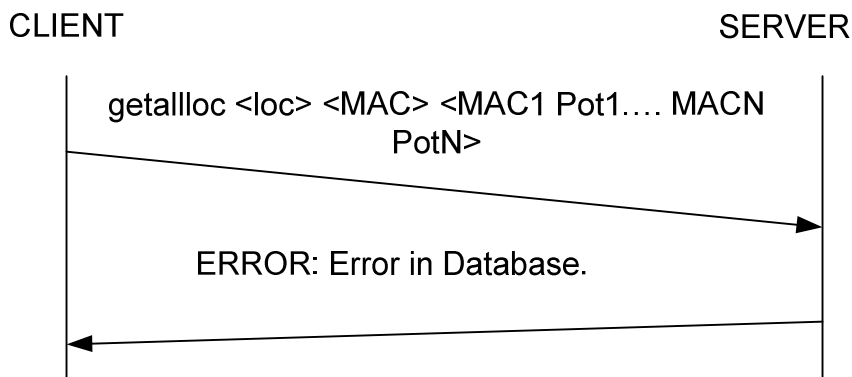
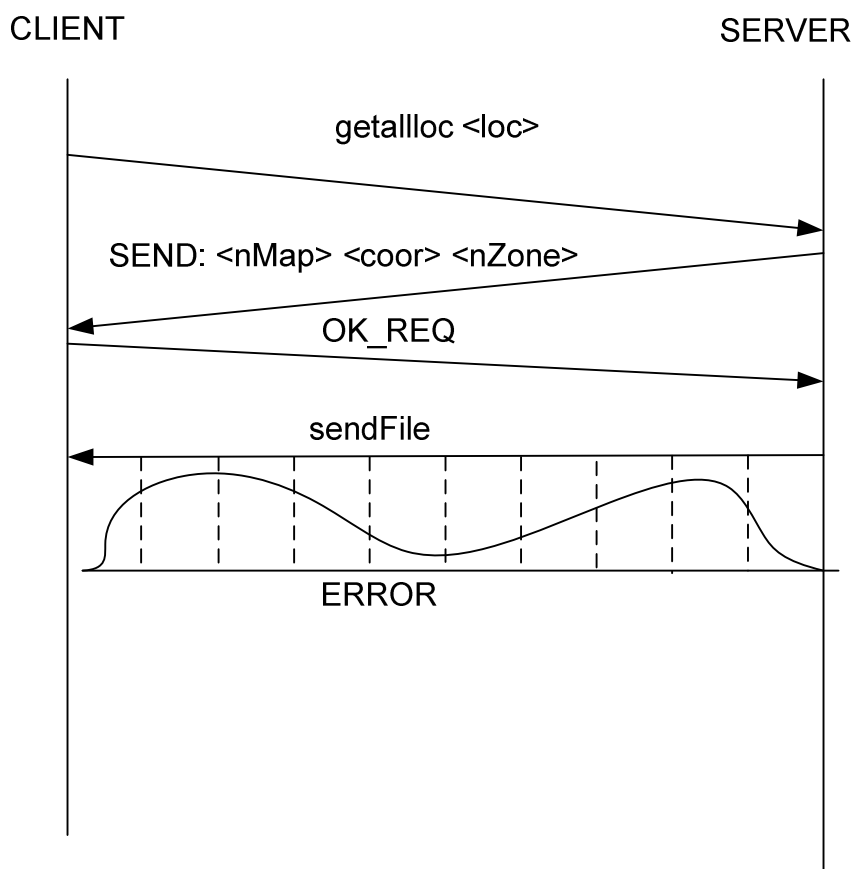


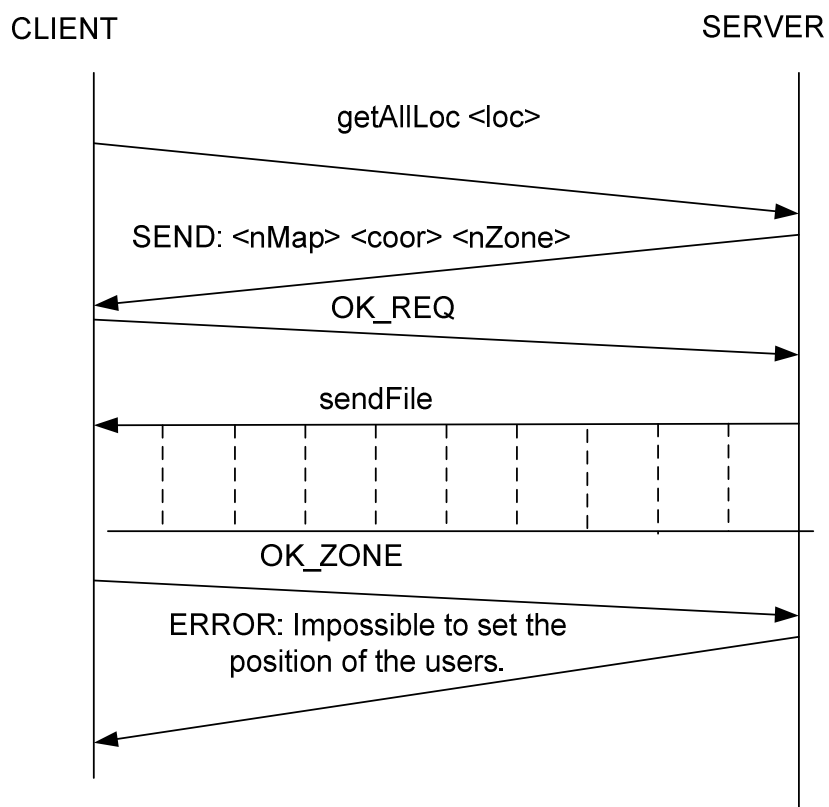
Fig 40: getAllLoc con errores.

En la figura (41) se puede apreciar un escenario de error de “getAllLoc”, debido a un problema en la recepción del mapa de la localización.



**Fig 41: getAllLoc con errores.**

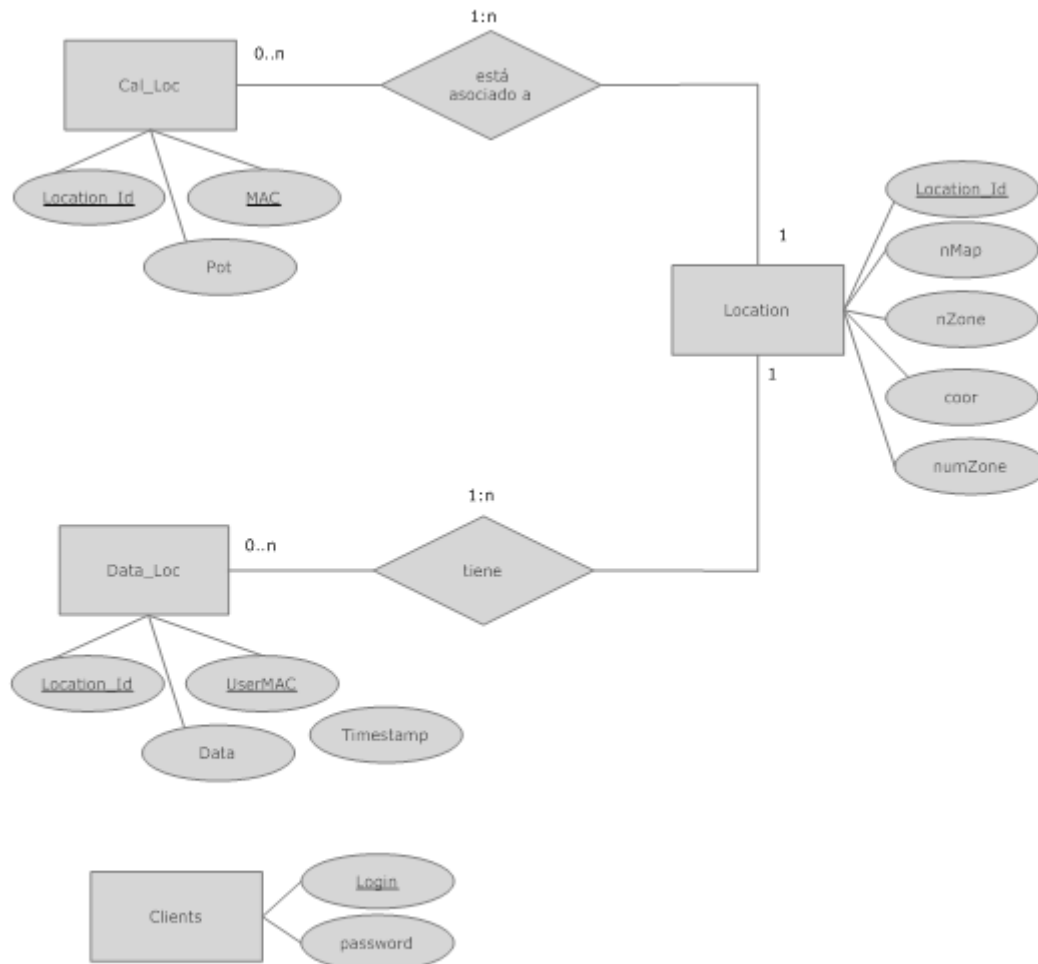
En la figura (42) se puede apreciar un escenario de error de “getAllLoc”, en el que no es posible posicionar a los usuarios dentro de la localización.



**Fig 42: getAllLoc con errores.**

#### 4.2.4 Base de Datos

El modelo conceptual de la base de datos usada puede verse en el siguiente diagrama:



**Fig 43: Modelo Conceptual de la Base de Datos.**

Por un lado, se encuentran los datos de autenticación del usuario del sistema, su login y password, posibilitando ofrecer control de acceso en el sistema, serán los atributos de la entidad “Clients”.

El núcleo de la base de datos es el objeto “Location”, que permite guardar distintos lugares donde localizar a usuarios, con su nombre que será identificador único, el nombre del mapa, y el número de zonas, puesto que para cada localización existe 1 a n zonas.

Cada localización puede tener 0 a n calibraciones, éstas se identificarán en la entidad ‘Cal\_loc’. Cada ‘Cal\_loc’ tendrá diversos APs identificados por su dirección MAC y la localización a la que pertenece su calibración de potencia “Location Id”, así como su Potencia de calibración.





Cada localización tendrá de 0 a n datos de los usuarios registrados en la localización “Data\_loc”, los usuarios se identificarán mediante su dirección MAC, ‘UserMAC’ y la localización a la que pertenecen, “Location Id”. De los usuarios a localizar del sistema es necesario conocer los datos registrados por su tarjeta inalámbrica “APs-Pot” que se denominará como “Data” y el momento en el que se registro dicho “Data” “TimeStamp”.

### ***4.3 Restricciones.***

- Las localizaciones tienen que ser únicas, no se permiten dos localizaciones con el mismo nombre.
- Las direcciones MAC se consideran identificador único de los dispositivos inalámbricos.
- Las potencias se almacenan con formato coma flotante (FLOAT).
- El nombre del mapa es variable por lo que hay que usar un VARCHAR.
- El login debe ser un identificador único, no se permiten 2 iguales

## ***5 Implementación***

### ***5.1 Decisiones Previas***

Antes de comenzar la implementación del sistema se tendrán que tomar una serie de decisiones sobre el Lenguaje de Programación a usar o el algoritmo de localización a implementar. Dichas decisiones deben basarse en los principios de **universalidad, eficiente, eficacia y efectividad**.

Así como se tendrán que cumplir los objetivos de este proyecto:

- Modularidad.
- Escalabilidad.
- Multiplataforma.
- Reusabilidad.
- Fácil despliegue.

#### ***5.1.1 El Lenguaje de Programación***

Para elegir el lenguaje de programación más adecuado para este sistema se tienen que tener en cuenta las siguientes consideraciones:

- Se busca un sistema totalmente modular y escalable.

Estas dos palabras traen a la memoria los lenguajes Orientados a Objetos, pues con ellos es mucho más fácil realizar esta modularidad, ya que son modulares en sí mismos y totalmente escalables. Esto lleva a la consideración de implementar el sistema en C++/C# o en Java.

- Se busca un sistema multiplataforma y de fácil despliegue y uso.

Teniendo en cuenta estas dos características que se deben cumplir en el sistema el lenguaje por el que hay que decantarse es Java, debido a sus características propias de:

- Portabilidad.
- Multipropósito.
- Seguridad
- Orientación a objetos puro.

Dentro del lenguaje Java se contemplan diversas opciones a elegir dentro de la arquitectura Cliente/Servidor:

- Servicios Web (con J2EE).
- J2SE

Se elige J2SE por ser el entorno de desarrollo estándar de Java para realizar cliente/servidor tradicional y por las implicaciones que conlleva tener que llamar a una aplicación externa para poder acceder a los datos de la tarjeta inalámbrica.

Dentro de J2SE se elige realizar el cliente gráfico con Java Swing, ya que proporciona:

- Multiplataforma 100%.
- API plenamente independiente del S.O.

### ***5.1.2 Algoritmo de Localización***

El sistema debe ser independiente del algoritmo de localización a utilizar. Pero se debe elegir uno para poder probar el sistema. Por este motivo se elegirá uno de los algoritmos más sencillos y con muy buenos resultados como es el “Vector de Potencias”.

### ***5.1.3 Base de datos***

Para acceder a bases de datos es mucho más útil usar un motor o servidor que hace las funciones de intérprete entre las aplicaciones y usuarios con las bases de datos.

Esta utilidad se traduce en ventajas, entre las que se pueden mencionar las siguientes:

- Acceso a las bases de datos de forma simultánea por varios usuarios y/o aplicaciones.
- Seguridad, en forma de permisos y privilegios, determinados usuarios tendrán permiso para consulta o modificación de determinadas tablas. Esto permite compartir datos sin que peligre la integridad de la base de datos o protegiendo determinados contenidos.
- Potencia: SQL es un lenguaje muy potente para consulta de bases de datos, usar un motor ahorra una enorme cantidad de trabajo.
- Portabilidad: SQL es también un lenguaje estandarizado, de modo que las consultas hechas usando SQL son fácilmente portables a otros sistemas y plataformas.

Debido estas características, se busca para este proyecto la instalación de un Servidor de Bases de Datos ‘open-source’, de fácil instalación y manejo, y que sea escalable y portable para cumplir con los objetivos del Proyecto.

Por lo que se elije MySQL, debido a sus ventajas:

- Escalabilidad: es posible manipular bases de datos enormes, del orden de seis mil tablas y alrededor de cincuenta millones de registros, y hasta 32 índices por tabla.
- MySQL está probado con multitud de compiladores y dispone de *APIs* para muchas plataformas diferentes.
- Conectividad: permite conexiones entre diferentes máquinas con distintos sistemas operativos.
- Es multihilo, con lo que puede beneficiarse de sistemas multiprocesador.
- Permite manejar multitud de tipos para columnas.
- Permite manejar registros de longitud fija o variable.
- Los drivers para su acceso mediante Java son fáciles de descargar y de conectar.

## 5.2 Diagrama de clases UML

### 5.2.1 Diagrama de clases UML del Servidor

En la siguiente figura (45), se observa el diagrama general de clases del servidor.

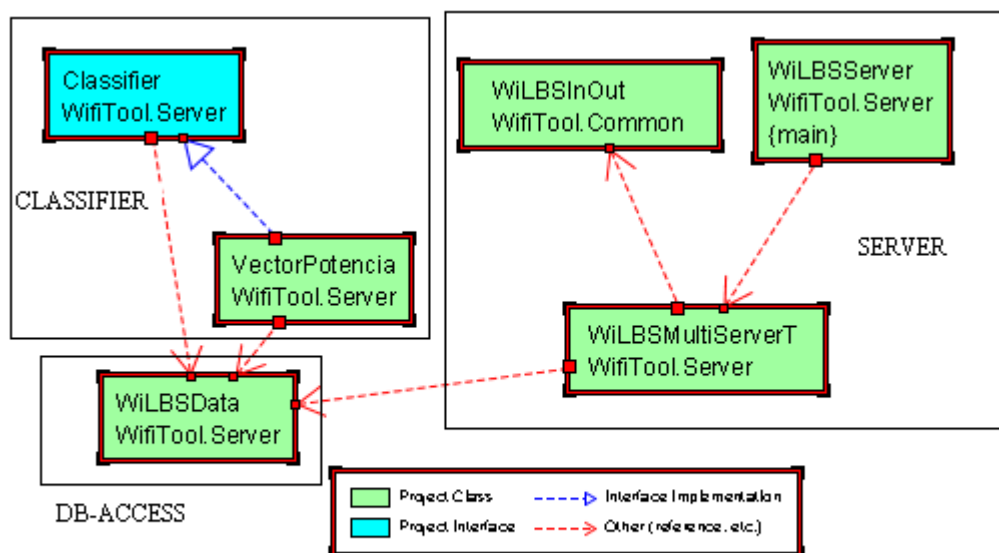


Fig 44: Diagrama de Clases del Servidor.

En él se pueden identificar los componentes en los que se divide el sistema:

- SERVER

Está compuesto por las clases `WiLBSServer` y `WiLBSMultiServerT`, donde la primera es la encargada de recepcionar las peticiones de los clientes e instanciar un nuevo hilo por cada cliente. `WiLBSMultiServerT` se encargará de controlar las peticiones de los clientes atendiendo al protocolo especificado en el Capítulo 4 de diseño.

Para tratar los datos de entrada/salida de los sockets y el envío controlado de ficheros binarios se ayudará de la clase auxiliar `WiLBSInOut`.

- DB ACCESS

Las funciones de este módulo las llevará a cabo la clase `WiLBSAccess`, que proporcionará al sistema el acceso a la base de datos.

Esta clase se conectará a la base de datos mediante un Front-End que accede al “middleware” específico para las bases de datos MySQL (**mysql-connector-java-5.1.6-bin.jar**), siguiendo el siguiente esquema:

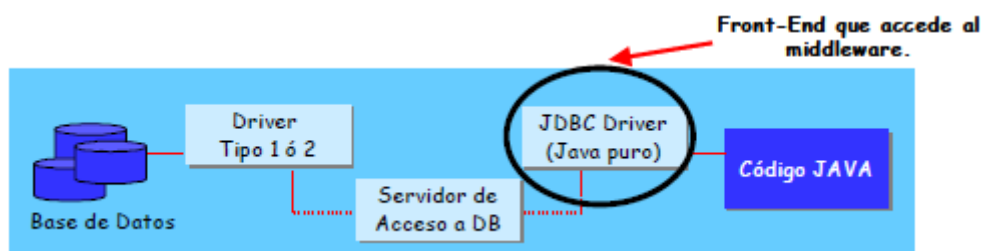



Fig 45: Conexión a Base de Datos.

- CLASSIFIER


Este módulo lo componen la interfaz `Classifier` y la clase `PowVector`. `Classifier` es la interfaz que permitirá el intercambio de los algoritmos de posicionamiento definiendo las funciones básicas para poder calcular el posicionamiento de un usuario. `VectorPotencia` es la clase que define el algoritmo “Vector Potencia” y que implementa la interfaz “`Classifier`”.

A continuación se muestra con detalle la implementación de las clases más relevantes del sistema.

 <b>WilBSMultiServerT</b>
<i>Attributes</i>
<code>private Socket socket = null</code>
<i>Operations</i>
<code>public WilBSMultiServerT( Socket newClient )</code> <code>public void run( )</code> <code>public void checkData( )</code> <code>public void delCal( DataOutputStream output, DataInputStream in, String loc )</code> <code>public void initCal( DataOutputStream output, DataInputStream in, String loc, String map, String coor )</code> <code>public void cal( DataOutputStream output, BufferedReader input, String loc, String zone, String data[0..*] )</code> <code>public void loc( DataOutputStream output, BufferedReader input, String loc, String MAC, String data[0..*] )</code> <code>public void getAllLoc( DataOutputStream output, BufferedReader input, String loc )</code> <code>public void setLoc( DataOutputStream output, BufferedReader input, String MAC, String data[0..*] )</code> <code>public void sendMsg( DataOutputStream out, String msg )</code> <code>public String receiveMsg( BufferedReader in )</code> <code>public void write( DataOutputStream dos, String s )</code>

**Fig 46: Clase WilBSMultiServerT**

WilBSMultiServerT, es el motor del sistema desde dónde se gestionan todas las llamadas al sistema y se coordina su funcionamiento. Se pueden apreciar los métodos de envío y recepción de mensajes a través de los sockets, así como los métodos que tratan los comandos del sistema ‘cal’, ‘loc’, ‘setLoc’, etc.

 <b>PowVector</b>
<i>Attributes</i>
<code>public Hashtable calDataInf</code> <code>public int nZone = 0</code>
<i>Operations</i>
<code>public PowVector( )</code> <code>public int max( double zone[0..*] )</code> <code>public String generateLetter( int i )</code>
<i>Operations Redefined From Classifier</i>
<code>public int getLoc( String location, WilBSDData data, String userData[0..*], int nZone )</code>

**Fig 47: Clase PowVector.**

PowVector implementa el algoritmo de Vector de Potencias. Se le proporcionan los datos de calibración a partir del elemento de data, acceso a la base de datos, la localización en la que hay que posicionar al dispositivo, los datos del dispositivo userData (array que contiene las parejas MAC Pot leídas por el dispositivo) y el número de zonas; y después de aplicar el algoritmo de Vector Potencias sobre los datos entregará un entero con el número de zona en el que se encuentra el dispositivo.

WiLBSData
<p><i>Attributes</i></p> <pre> private Statement stmt = null private String user_login private String user_password private boolean connected = false private Connection conn </pre>
<p><i>Operations</i></p> <pre> public WiLBSData( ) public boolean connect( ) public void disconnect( ) public Statement getStatement( ) public void setStatement( Statement s ) public void setUserLogin( String u_l ) public void setUserPassword( String u_p ) public String getUserLogin( ) public String getUserPassword( ) public boolean isValid( ) public boolean password( ) public boolean isConnected( ) public boolean addClient( String new_login, String new_pass ) public boolean locationExists( String loc ) public boolean setLocation( String loc, String map, String coor ) public boolean delLocation( String loc ) private int calculateZone( String coor ) public String getLocations( ) public String getMACs( String location ) public String searchLoc( String mac ) public String getMap( String Location ) public String getCoor( String Location ) public Float[0..*] getPow( String loc, String MAC, int nZone ) public boolean checkTable( String table ) public boolean createTable( String name, int numColumns ) public boolean createTable( String name ) public Timestamp getTimeStamp( String location, String MAC ) public void checkTimeStamp( String location, Timestamp sqlDate ) public int nZone( String loc ) public boolean insertData( String loc, String zone, String MAC, String Pot ) public boolean insertData( String loc, String mac, String data, Timestamp time ) public boolean dropTable( String name ) </pre>

Fig 48: WiLBSData clase.

WiLBSData se encarga del acceso a los campos y tablas de la base de datos.



## 5.2.2 Diagrama de clases UML del Cliente.

Para realizar las pruebas del sistema se implementan tres clientes:

- WiLBSGUI, es un cliente gráfico que proporciona acceso al sistema de forma gráfica a los tres tipos de usuario del sistema Gauge, Mobile/User y Monitor.
- User, es un cliente de activa la función de “localizable” en el sistema.
- WiLBSClient, es un cliente en modo texto para probar el protocolo de comunicación mediante el envío de comandos al servidor.

En la figura (49), se observa el diagrama general de clases del cliente gráfico para los usuarios “Gauge”, “Mobile/User” y “Monitor”.

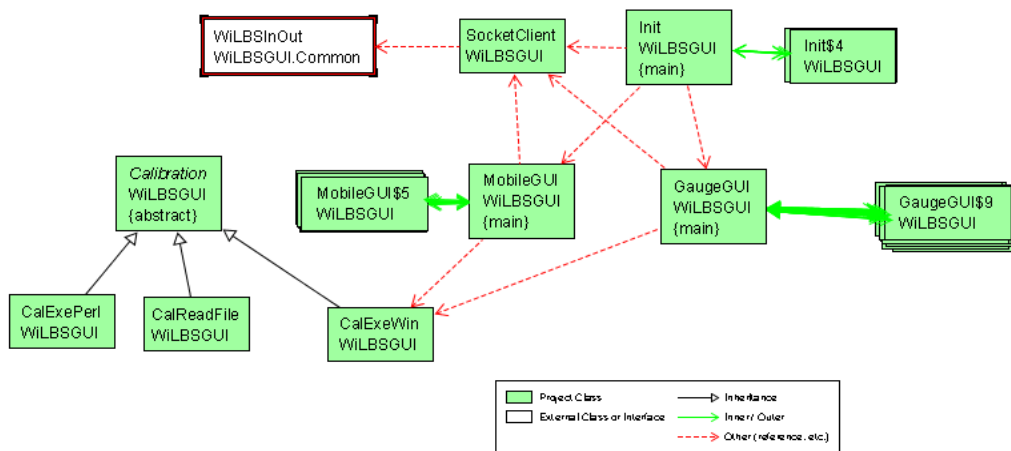


Fig 49: Diagrama de clases del Cliente Gráfico.

En la figura (50), se observa el diagrama general de clases del cliente “Mobile/User” de activación de la función de estar localizable.

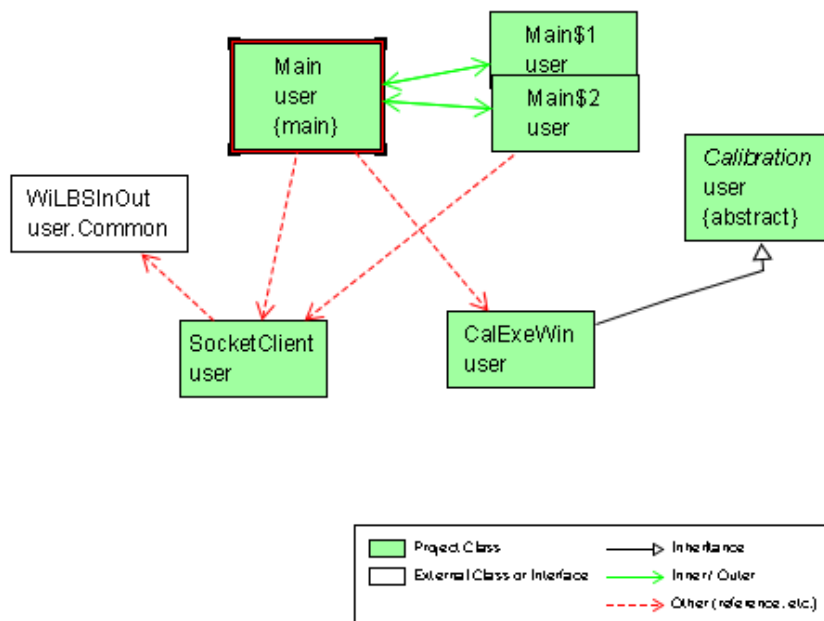




Fig 50: Diagrama de clases del Cliente User.

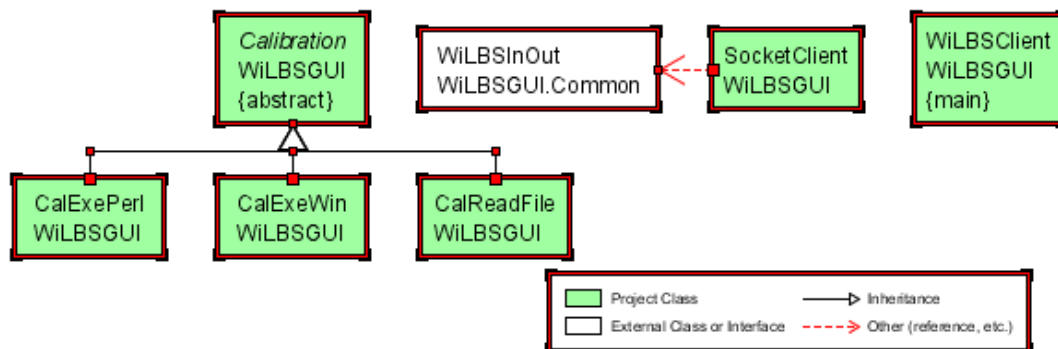


Fig 51: Diagrama de clases del Cliente Texto.

En ellos se pueden identificar los dos componentes en los que debe modularizarse el cliente:

- PRESENTATION

Este módulo lo componen las siguientes clases:

- SocketClient, que será el subsistema encargado de las comunicaciones con el sistema. En el está implementado el protocolo propietario de comunicación sobre sockets.
- WiLBSInOut, clase auxiliar para el control de entrada/salida de los datos de los sockets y del envío controlado de ficheros binarios.
- Init, GaugeGui, MobileGui y MonitorGui, en el caso del cliente gráfico para Gauge, Mobile/User y Monitor, proporcionan la “Vista” del sistema. Encargándose del diseño gráfico, Init para la pantalla principal y acceso al servidor, GaugeGui para la pantalla correspondiente al cliente Gauge, MobileGui para el cliente Mobile/User, y MonitorGui para las funcionalidades del cliente Monitor.
- Main, en el caso del cliente de activación de la función de “localizable” de un Mobile/User, será la encargada de la “Vista” de este cliente. En este caso será un icono en la barra de herramientas que se mostrará mientras este activa esta función.
- WiLBSClient, el cliente modo texto encargado de leer por línea de comandos las peticiones que se le quieren solicitar al sistema.

- WIFI-ACCESS

En este caso se tienen las siguientes clases:

- Calibration, clase abstracta que permite intercambiar el módulo de acceso en función del S.O. que se posea o del tipo de Acceso a la Tarjeta inalámbrica que se tenga:
  - CalExePerl, proporciona la invocación a una función en “perl” de “wifipos.pl” en la que se puede recuperar los datos requeridos de una tarjeta inalámbrica en un S.O. Linux/Unix.
  - CalExeWin, que invoca a “wifipos.exe” en S.O. Microsoft Windows, acceso mediante el protocolo NDIS a la tarjeta, para recuperar los datos requeridos de la tarjeta inalámbrica.
  - CalReadFile, accede a un fichero en los que se hayan guardado los datos de la calibración de una zona con el formato de datos requerido “Zone MAC MAC1 Pot1 ... MACN PotN”.

### 5.3 Base de datos.

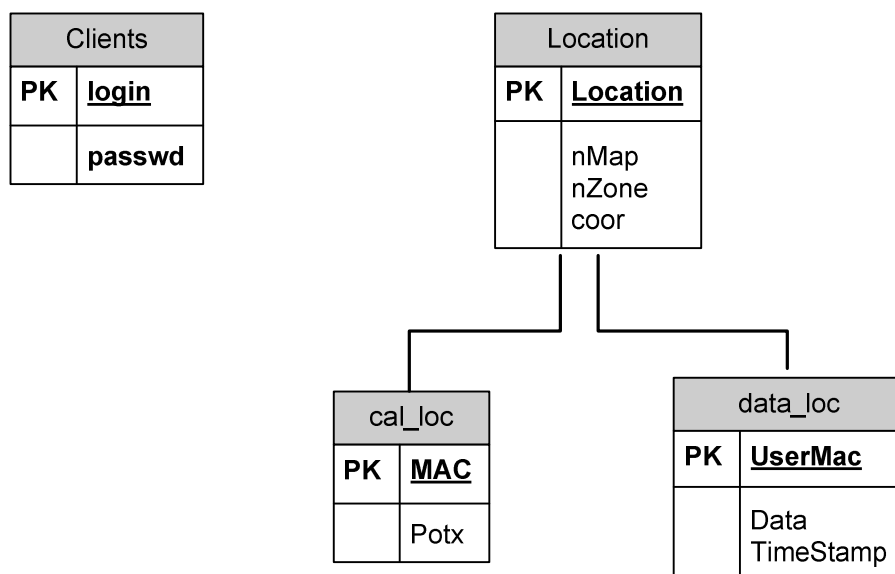


Fig 52: Base de datos

En la base de datos, se definen los siguientes tipos de tablas:

- Clients

Su clave primaria será el campo ‘login’ que es el identificador único de usuario del sistema, puede tomar cualquier valor alfanumérico “pepe”, “luis”, “root”.

Cada usuario poseerá una contraseña para acceder al sistema que se almacenará en el campo “passwd”, que soportará valores alfanuméricos.

- **Locations**

Cada tabla “locations”, se identificará por una clave primaria, la cual será un nombre único “location”, no nulo. El campo location puede ser “Planta 1”, “Torres\_Quevedo”...

Los datos que tiene que tener una localización serán:

- El nombre del mapa de la localización, nMap, que es un VARCHAR(255).
- El número de zonas en el que se divide la localización, nZone, será un valor entero.
- Las coordenadas de la localización, que se almacenaran como un String.

- **Cal\_loc**

En esta tabla se almacenaran las calibraciones de una localización. Como clave primaria tiene la MAC address de los APs visibles, y como ‘foreign key’ el nombre de la localización.

De cada MAC se debe almacenar la Potencia en cada zona, por lo que tendremos tantas potencias de esa MAC como zonas haya en la localización. Las potencias deben almacenarse como un Float.

- **Data\_loc**

En esta tabla se almacenan los datos de los usuarios registrados en una localización. Como identificador se tiene a la dirección MAC del usuario, ‘UserMAC’, y como identificador secundario la localización en la que se encuentra, ‘location’.

De cada usuario se almacenan los datos que recoge, parejas de MAC y potencias, en el campo ‘data’ y el momento en se ha recibido información de ese usuario, ‘timestamp’.

## **5.4 Restricciones.**

- El fichero del mapa no se puede encontrar en directorios que contengan espacios.

## ***6 Pruebas y Resultados***

En esta sección se va a proceder al análisis de los tipos de pruebas ejecutados y sus resultados. Así, se comienza con una descripción somera del entorno de pruebas necesario, para pasar a analizar las pruebas de GUI y usabilidad de los distintos clientes, acabando con detalles sobre la clasificación en localizaciones de distintos clientes.

### ***6.1 Entorno de Pruebas***

Las pruebas del sistema se han realizado en un entorno conocido, y con gran cobertura inalámbrica: “La planta Primera del Edificio Torres Quevedo de la Universidad Carlos III de Madrid”.

El sistema presentado en este proyecto consta, como ya hemos visto, de uno o varios clientes y de un servidor. Para probarlo, pues, es necesario disponer de distintas plataformas en las que éstos se puedan ejecutar. Además, la plataforma donde se ejecute el cliente móvil ha de constar, lógicamente, de una o varias tarjetas inalámbricas Wi-Fi.

Afortunadamente, la presencia de tarjetas inalámbricas es un hecho común en casi cualquier ordenador personal actual, y, por tanto, no resulta sorprendente que cualquiera de éstos haya servido como entorno de pruebas del cliente.

Por otro lado, el servidor no necesita tarjeta inalámbrica pero si de la presencia de algunos servicios, como la base de datos MySQL, en él. Al ser ésta de descarga e instalación gratuita, y no imponer excesivas cargas en los requerimientos de sistema, no ha resultado ningún problema disponer de varios ordenadores que pudieran actuar como servidor si fuera necesario.

Para que las pruebas puedan realizarse en distintos puntos y poder comparar los resultados obtenidos en ellos, amén de probar la correcta clasificación en una zona dada, es necesario que el cliente sea móvil – por ello, entre otros clientes, se ha usado un portátil para realizar las pruebas en distintos entornos físicos.

La siguiente figura muestra el mapa ejemplo de la localización elegida.

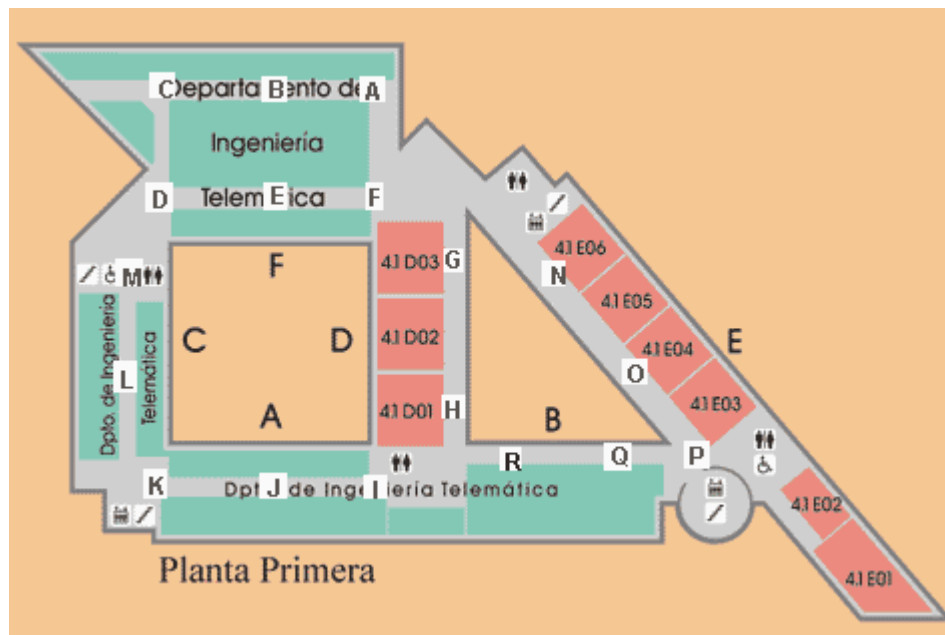


Fig 53: Ejemplo de Mapa de Localización.

## 6.2 Cliente de Pruebas

Se describen, a continuación, los distintos clientes usados y sus pruebas de funcionalidad y GUI. Al probar el correcto funcionamiento de la GUI y su correcta interacción con las distintas opciones del servidor, se están probando, implícitamente, los casos de uso definidos en las secciones anteriores.

### 6.2.1 Cliente Texto

Este cliente permite introducir directamente comandos del protocolo que espera recibir el servidor, y, sirve, por tanto, como una forma de probar el correcto funcionamiento del protocolo. Realiza, por tanto, la misma funcionalidad que en otros protocolos de comunicación puede realizar telnet – ofrecer un cliente sencillo y textual que permita, mediante la escritura directa de comandos, verificar el correcto funcionamiento del protocolo subyacente. No tiene ninguna utilidad adicional, y, aunque puede llegar a ser completamente funcional si se introducen los comandos en el formato correcto, resulta incómodo y arcano para un usuario estándar.

## 6.2.2 Cliente Gráfico

En las pruebas con este cliente, se irá pantalla a pantalla probando su funcionalidad para posteriormente proceder a la comprobación del funcionamiento del algoritmo de localización

El cliente gráfico se ha diseñado para ser fácilmente empleado por cualquier tipo de usuario, independientemente de sus conocimientos de comunicaciones inalámbricas.

El sistema arranca con una pantalla en la que es posible introducir el tipo de cliente que se desea usar, el nombre de usuario y su contraseña. Si esta contraseña y nombre de usuario son aceptados por el servidor (que se determina mediante su IP o URL), el cliente pasará a mostrar las siguientes opciones, que dependen, del tipo de cliente que se use en la conexión.

En esta pantalla inicial resulta útil probar que ocurre cuando se introducen *datos incorrectos* (IP que no corresponda a un servidor, nombre de usuario inexistente, clave incorrecta) o, directamente, *inválidos* (Error de usuario/password). También mostrará un mensaje de error en el caso de dejar algún campo en blanco.



Fig 54: Pantalla Inicial.

En ella se introducen los datos de conexión al sistema.

A continuación se muestra una prueba introduciendo datos correctamente al sistema.



Fig 55: Pantalla inicial con datos correctos.

Y otra prueba introduciendo un usuario no válido en el sistema.



Fig 56: Pantalla inicial – Error en el usuario o en el password.

Al pasar estas comprobaciones, se ha verificado el correcto funcionamiento del **caso de uso Connect**.

### 6.2.3 Gauge

Suponiendo que el usuario ha escogido entrar en el sistema como un cliente de tipo Gauge, lo primero que encuentra es una pestaña con todas las localizaciones existentes.

Las pruebas que se pueden realizar en este caso consisten en la correcta visualización de todas las entradas disponibles en la base de datos. El cliente pasa correctamente esta prueba.

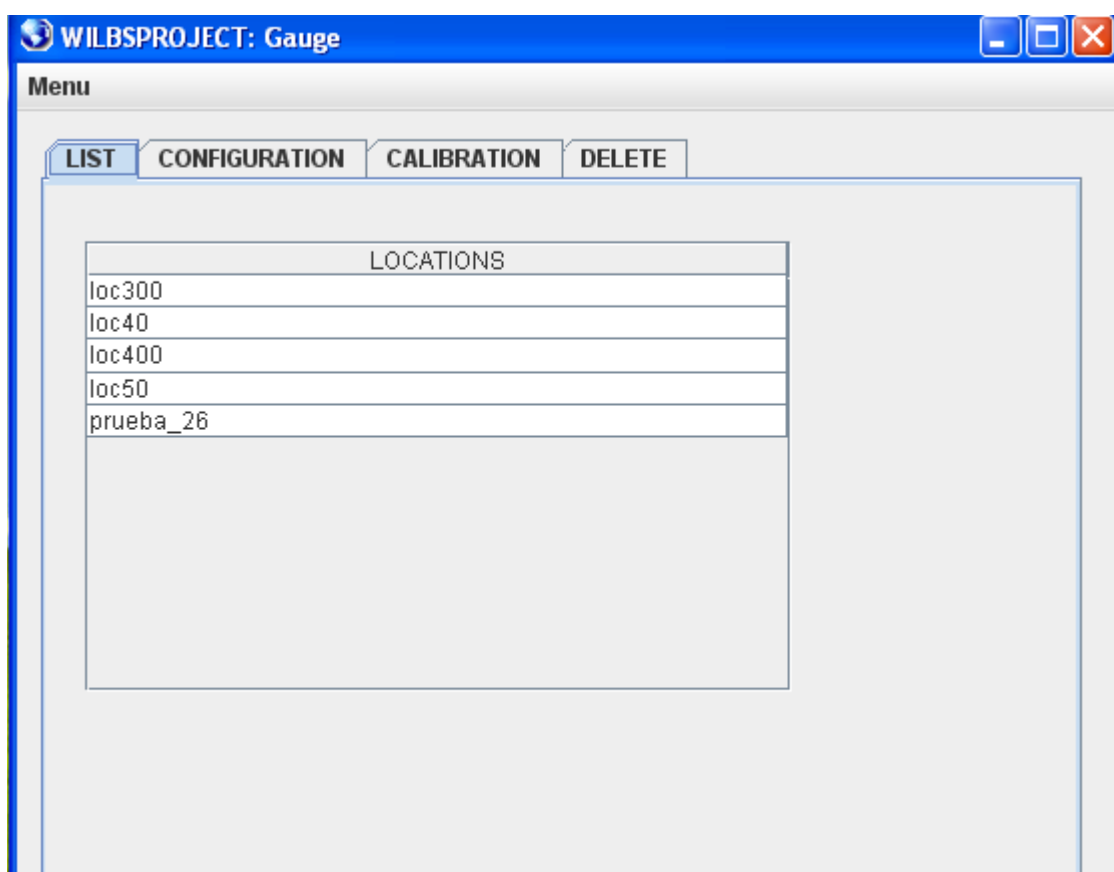


Fig 57: Pantalla de listado de localizaciones.

Al pasar estas comprobaciones, se ha verificado el correcto funcionamiento del **caso de uso List**.

La segunda pestaña disponible en el modo Gauge permite añadir una nueva localización. Para ello, la interfaz acepta que se indique un mapa (a través de un fichero con una imagen, bien .jpg o .gif), y que se añadan y borren zonas con los botones "Add Zone" y "Delete Zone". Cuando todos los datos han sido introducidos, puede enviarse esta información al servidor mediante el botón "Add Loc Conf".

Las pruebas disponibles en este caso son variadas: subir distintos tipos de mapas, definir distintas zonas, y comprobar que se reflejan correctamente sobre el mapa; borrar zonas y comprobar que estas también se reflejan correctamente, subir una localización al servidor y verificar que la localización se queda guardada, y, por último, intentar subir



una localización incorrecta y verificar que se rechaza (Por ejemplo una localización existente).

En su estado actual, el cliente pasa, evidentemente, todas estas verificaciones.

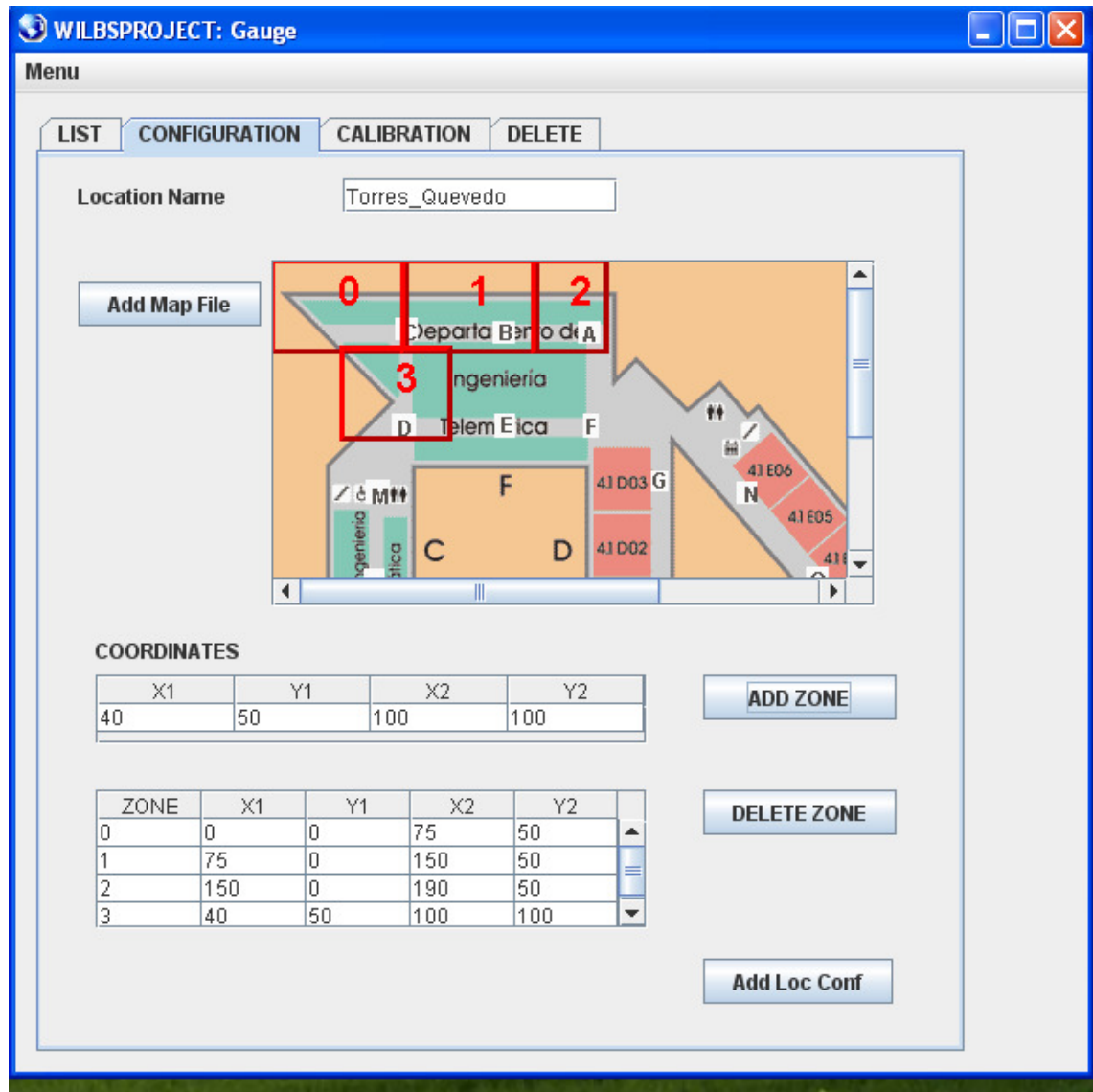


Fig 58: Pantalla de configuración de las localizaciones

El correcto funcionamiento de estas pruebas verifica el caso de uso Initial.

El cliente también permite añadir datos de calibración, para cada zona. Para ello, dispone de una pestaña en la que se permite seleccionar la localización y zona actual del terminal, para, posteriormente, introducir dicha información en la base de datos a través del botón "CALIBRATE".

Las pruebas que se pueden realizar en esta pestaña son menos variadas que en la anterior: consiste, básicamente, en comprobar que todas las localizaciones están disponibles y que todas las zonas para una localización están disponibles, además de

comprobar que los datos efectivamente se insertan en la base de datos, como puede verse en la siguiente figura.

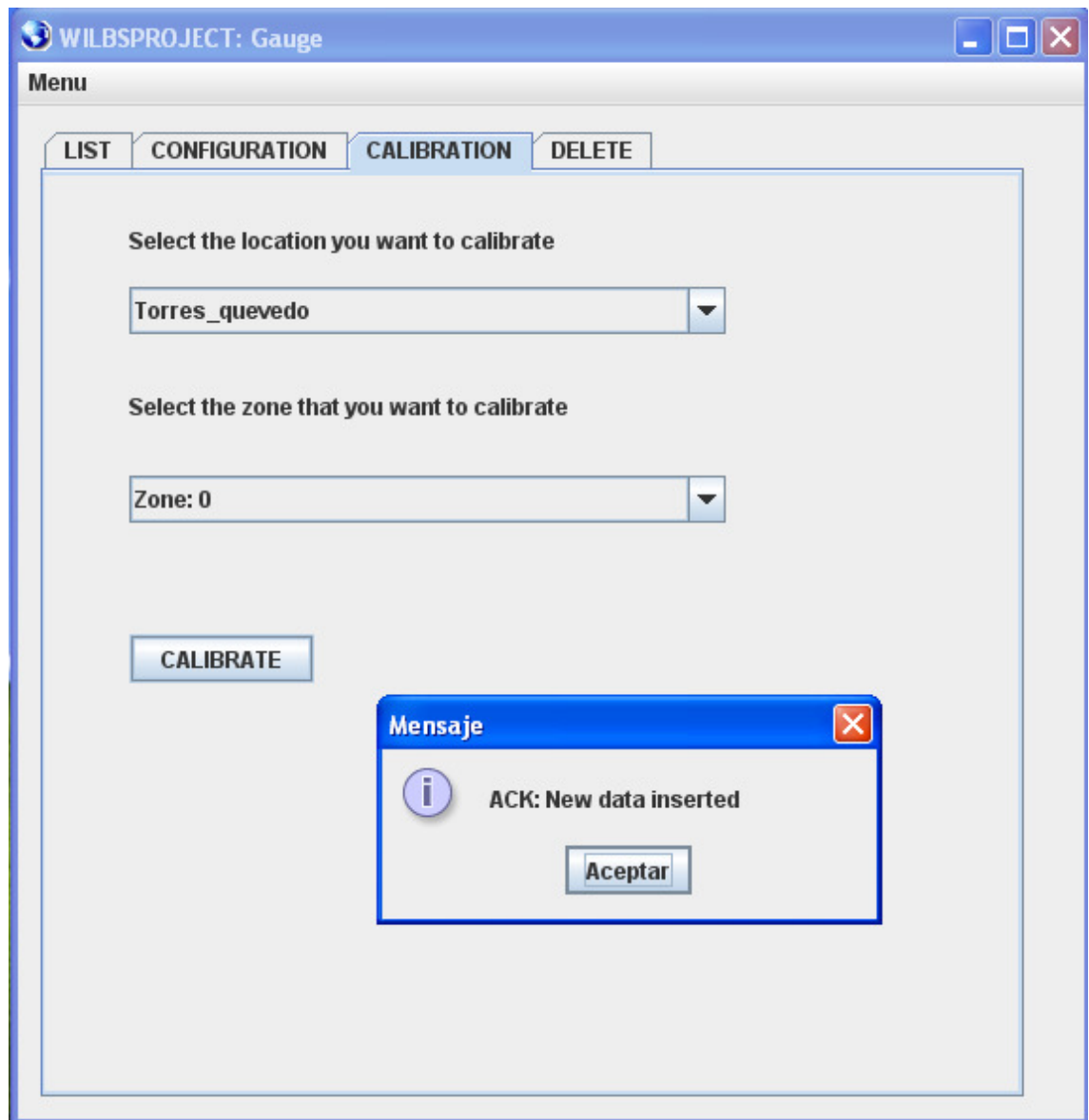
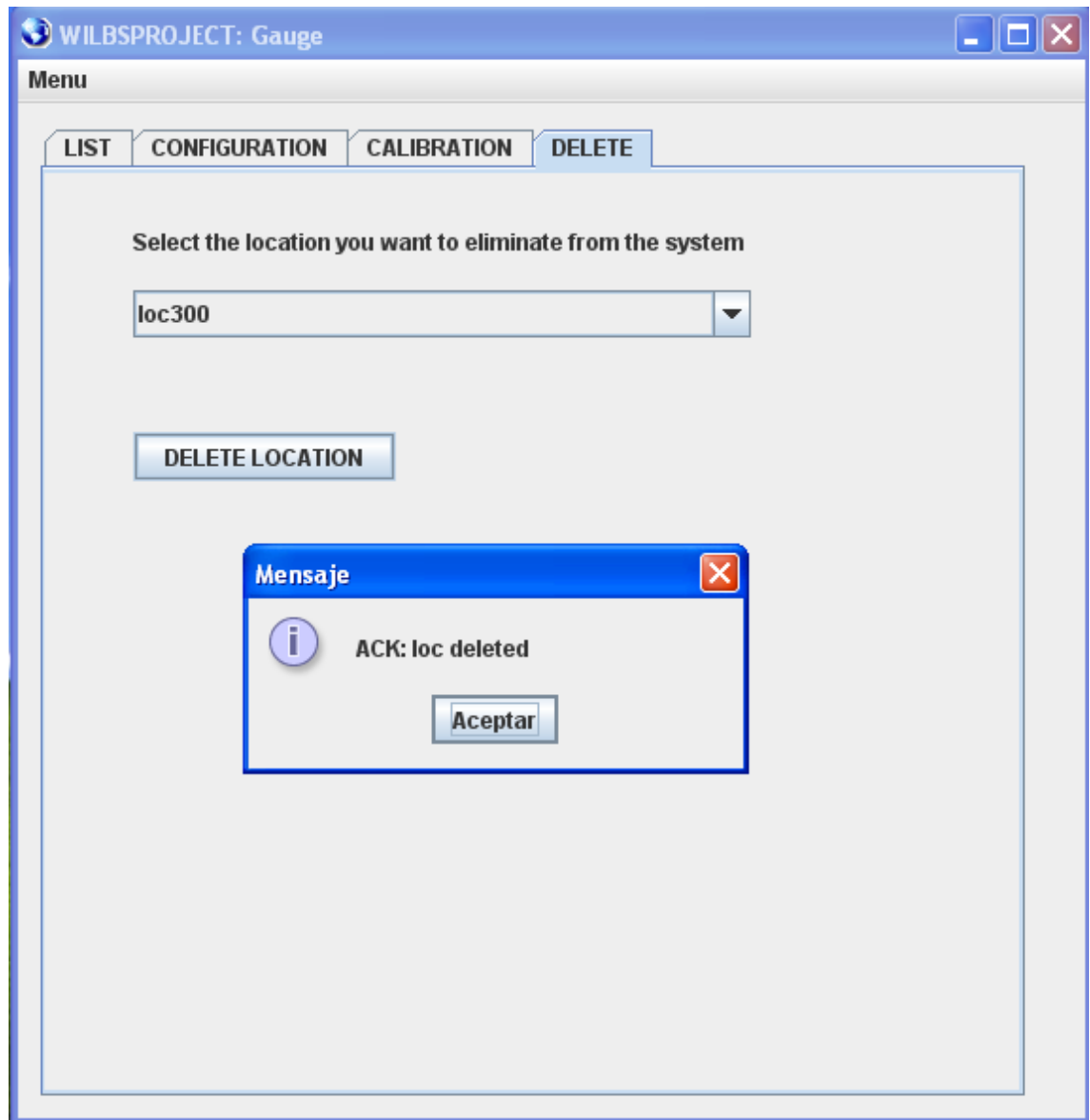


Fig 59: Pantalla de calibración de la localización.

Con esta pestaña se comprueba el funcionamiento del **caso de uso Cal.**

Por último, para permitir el borrado de una localización ya existente, el sistema dispone de una cuarta pestaña, que sólo permite seleccionar la localización a borrar. Las pruebas que pueden realizarse sobre esta pestaña son también muy limitadas, y consisten, nuevamente, en verificar que todas las localizaciones aparecen y que cualquiera de ellas puede ser borrada.



**Fig 60: Pantalla de borrado de la localización.**

Evidentemente este caso de prueba se corresponde con **el caso de uso Delcal**.

## 6.2.4 User/Mobile

El cliente User/Mobile está orientado a permitir descubrir a un usuario en que zona se encuentra. Para ello, se le pide que seleccione su localización y pulse el botón “LOCATE”.

En este caso, las pruebas son muy simples y consisten, básicamente, en comprobar que la localización es correctamente reflejada en el cliente. Se verifica así el **caso de uso Loc.**

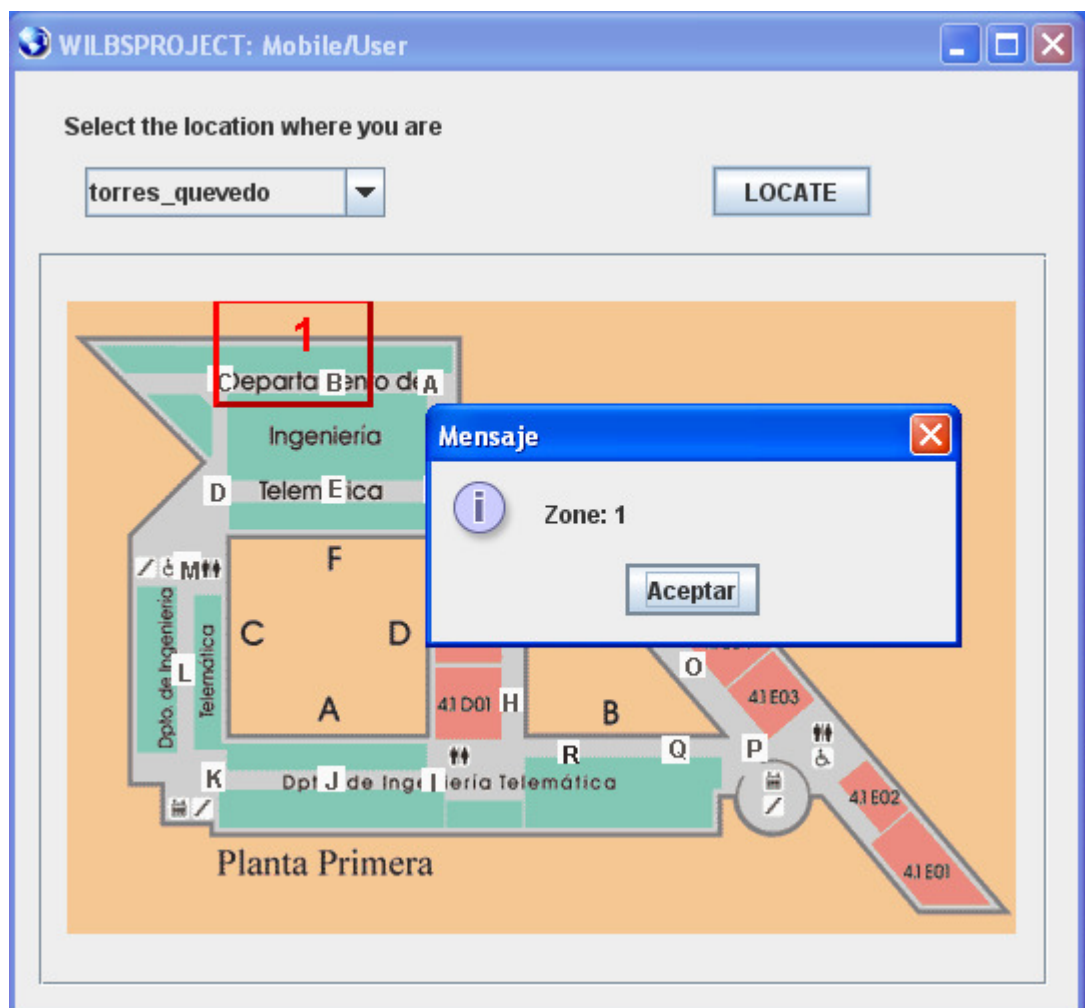


Fig 61: Pantalla de Posicionamiento del usuario.

Este usuario también puede activar la opción de ser localizado. Para ello ejecutará el cliente User, que se conectará con el sistema y comenzará a enviarle los datos al sistema cada cierto tiempo (por defecto 30 segundos). Mientras este cliente este activo el usuario podrá ver en la barra de tareas el siguiente icono.



Fig 62: Icono de la aplicación.

Para desactivarlo deberá pulsar sobre el icono el botón derecho del ratón y pulsar sobre la palabra “Exit”.

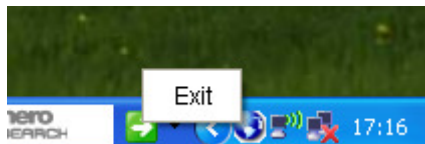


Fig 63: User- desactivar opción de “localizable”

### 6.2.5 Monitor

El cliente Monitor está orientado a permitir descubrir donde se encuentran los usuarios de una localización. Para ello, se le pide que seleccione la localización y pulse el botón “LOCATE”.

En este caso, las pruebas son muy simples y consisten, básicamente, en comprobar que la localización de cada usuario es correctamente reflejada en la pantalla. Se verifica así el **caso de uso getAllLoc**.

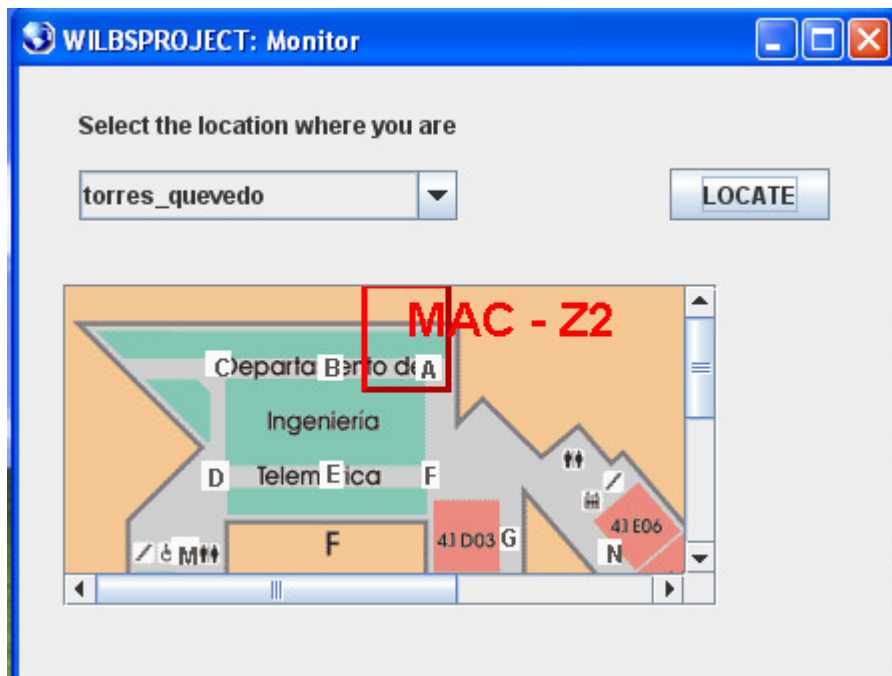


Fig 64: Monitor – Localización de usuarios

### ***6.3 Resultados de calibración***

Las pruebas de la exactitud calibrando han sido realizadas principalmente en el entorno geográfico de la planta primera del edificio Torres Quevedo de la Universidad Carlos III de Madrid, y han consistido en las siguientes pruebas, diseñadas para encontrar los límites de la localización:

- 1) Definir dos grandes zonas y comprobar que en el centro de ambas zonas la localización era correcta.
- 2) En el mismo escenario, ir acercándose hasta el borde de la zona y comprobar cuando se produce una localización errónea.
- 3) Definir tres zonas y repetir los escenarios anteriormente mencionados
- 4) Ir aumentando el número de zonas hasta encontrar un número máximo que puede ser definido, a partir del cual incluso en el centro exacto de la zona no se produce una localización correcta.

Las pruebas han dado muy buenos resultados, con un 0% de fallos en los 5 primeros casos, aumentando el error en según se disminuía el tamaño de las zonas en las que se dividía la localización. También se ha podido apreciar como era más exacta dicha localización cuando se habían tomado más calibraciones sobre esa zona o cuando la calibración se había realizado con menos dispositivos móviles en la localización.

## ***7 Conclusiones y Líneas Futuras***

### ***7.1 Conclusiones***

En este Proyecto se ha realizado un prototipo de sistema de posicionamiento en interiores para aprovechar el gran despliegue de redes de comunicaciones inalámbricas, en concreto el estándar WiFi.

Actualmente, como se ha mostrado en la exposición del estado del arte, existen una multitud de sistemas desarrollados tanto en el entorno académico como en el comercial que intentan dar una solución al posicionamiento de terminales móviles que usan sistemas de comunicación inalámbricos. En opinión de los autores, estos sistemas son cerrados e independientes entre sí, y no permiten modificaciones necesarias para la investigación de nuevas soluciones. Por tanto, la motivación de este Proyecto es cubrir la necesidad que existe de una plataforma totalmente abierta y modular, que permita fácilmente la evaluación de diferentes algoritmos de posicionamiento, y de manera que pueda evolucionar haciendo uso de las nuevas tecnologías.

A la hora del diseño del sistema, era deseable que fuera multiplataforma, y que funcionara en los sistemas operativos más utilizados, Microsoft Windows y Linux. También tendría que ser independiente de la topología de la red de comunicaciones subyacente. Esta razón ha motivado la utilización del lenguaje de programación Java, por ser un sistema orientado a objetos puro, ampliamente difundido, y capaz de ejecutarse con independencia de la máquina y de su sistema operativo.

La arquitectura es del tipo Cliente/Servidor, a través de la cual distintos dispositivos móviles podrán conocer su localización. Por tanto, se ha creado la infraestructura necesaria para el desarrollo de aplicaciones que ofrezcan los servicios basados en localización: cada una de estas aplicaciones le basta con comportarse como clientes en el esquema desarrollado en este Proyecto para poder ofrecer servicios a sus usuarios. La comunicación entre los Clientes y el Servidor se realizan mediante sockets.

El Servidor es el encargado de la mayor parte de la lógica del sistema, como la gestión de las localizaciones, de los diferentes clientes, de comunicarse con la base de datos MySQL, cuyo diseño se ha realizado en este Proyecto, donde se almacena todos los datos relacionados con la aplicación, así como de generar el cálculo de las posiciones pedidas por los clientes. Cualquier cambio o mejora ha de hacerse sólo en el Servidor, lo cual facilita la gestión del sistema.

Por otra parte, los Clientes se han diseñado con la máxima simplicidad posible, teniendo en cuenta que la mayor parte de dispositivos móviles están limitados en cuanto a su

capacidad de cómputo. A modo de ejemplo, en este Proyecto se han desarrollado dos tipos de Clientes, uno en modo texto, y otro en modo gráfico.

Uno de los mayores problemas encontrados en la realización de este Proyecto ha sido el acceso al driver de la tarjeta inalámbrica del sistema operativo necesario para acceder a las medidas de potencia. Las restricciones de seguridad de Java impedían acceder directamente al hardware. Esto ha impedido que los clientes fueran “Clientes Ligeros”, ya que es necesaria la ejecución de llamadas a funciones externas (dependientes del sistema operativo) para obtener dichas medidas.

La aplicación está perfectamente documentada, con una interfaz clara que describe el intercambio de peticiones y respuestas entre Cliente y Servidor, lo cual hace sencillo el desarrollo de nuevos Clientes que aprovechen la infraestructura desplegada.

Todas las decisiones de diseño han permitido que esta infraestructura Cliente/Servidor no se encuentre limitada para evolucionar en el futuro, permitiendo que esta plataforma pueda adaptarse a las necesidades de los servicios de localización basados en redes de comunicaciones, que se vislumbra como uno de los sectores más prometedores en el mercado de las telecomunicaciones.

## ***7.2 Líneas Futuras***

Como se ha destacado a lo largo de esta Memoria, este Proyecto Fin de Carrera se ha desarrollado pensando en el futuro, ya que su objetivo es servir como herramienta a posteriores desarrollos.

Por tanto, la principal línea futura de investigación es el despliegue de la herramienta en entornos reales de trabajo y usarla como plataforma de pruebas de nuevos algoritmos de localización que usen las medidas de potencia para dar una posición de los terminales móviles cada vez con más precisión. Esto es de gran interés debido a que el sistema principal de navegación en la actualidad, el GPS, no funciona en el interior de edificios, por lo que existe un gran empuje por parte de la industria y del mundo de la investigación en conseguir una integración entre el sistema de GPS con otro que funcione en interiores para dar un servicio continuo.

Por otra parte, una vez que se tiene la infraestructura de localización que se ha desarrollado en este Proyecto, el siguiente paso es realizar los servicios que hagan uso de ella y le aporten valor al usuario.

Por último, debido al diseño modular que se ha seguido, no hay nada que impida utilizar la aplicación aplicándola a otros sistemas de comunicación. Igual que funciona usando WiFi para entornos de red local inalámbrica en interiores, un trabajo futuro sería ampliarla para que utilice redes WiMax en entorno de redes metropolitanas, o cualquier otro sistema de comunicaciones.





Para que sea posible el desarrollo de aplicaciones realmente portables, sería necesario la creación de una API de posicionamiento que permitiera el acceso a las medidas de potencia de una manera única y estándar, con independencia del tipo de terminal, de la red y del sistema operativo. De esta manera se evitarían los problemas detectados durante el transcurso de este Proyecto.

Otra de las posibles vías de ampliación de este Proyecto sería la realización de clientes para plataformas móviles como pueden ser Symbian o J2ME. No resultaría extremadamente difícil puesto que las últimas versiones de ambos soportan la ejecución de aplicaciones que se comuniquen mediante sockets; por lo tanto, únicamente se necesitaría la implementación de clientes sobre estas plataformas con acceso WiFi que implementen el protocolo de comunicación detallado en esta memoria.

## 8 Bibliografía y Referencias

- [1] Accuracy Enhancement of an Indoor ANN-based Fingerprinting Location System Using Particle Filtering and a Low-cost Sensor. Salim OUTEMZABET. Chahé NERGUIZIAN. École Polytechnique de Montréal, Canada
- [2] Advanced Wavelan Positioning System.  
<http://www.cdt.luth.se/projects/alipes/public/Reports/2001-05-23%20Advanced%20WaveLAN%20Positioning%20System/2001-05-23%20Advanced%20WaveLAN%20Positioning%20System.pdf>
- [3] *Amulet: Approximate Mobile User Location Tracking System*, Blake M. Harris, University of Rochester, 2002.
- [4] An indoor location scheme based on wireless local area Networks.  
<http://ieeexplore.ieee.org/Xplore/login.jsp?url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F9640%2F30469%2F01405248.pdf%3Farnumber%3D1405248&authDecision=-203>
- [5] Conceptos Básicos de GPS. Instituto de Agrimensura - Uruguay  
[www.fing.edu.uy/ia/deptogeo/elemtopo/GPS.ppt](http://www.fing.edu.uy/ia/deptogeo/elemtopo/GPS.ppt)
- [6] Ekahau: <http://www.ekahau.com/>
- [7] Ekahau Ranks Highest in Evaluation of Five Leading WiFi RTLS Solutions.  
[http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=4476251](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4476251)
- [8] Estudio de disponibilidad de medidas de localización en redes celulares urbanas. I. Martin-Escalona, F. Barcelo-Arroyo.
- [9] Galileo [http://www.esa.int/esaNA/SEMJQSXEM4E\\_galileo\\_0.html](http://www.esa.int/esaNA/SEMJQSXEM4E_galileo_0.html)
- [10] Google Latitude  
<http://www.google.com/support/mobile/bin/answer.py?answer=105928&cbid=nrqzov39zpbp&src=cb&lev=answer>
- [11] Herecast: [www.herecast.com](http://www.herecast.com)
- [12] ILS (Indoor Location Systems) Sistemas de Localización en Interiores. Raúl Sánchez Vítóres. Director de proyectos de MIPSAs
- [13] Innovative Cisco Wireless Location Appliance: The Right Tools to Visualize and Control RF Domains. May 4, 2005. By Jenny Carless, News@Cisco  
[http://newsroom.cisco.com/dlls/2005/hd\\_050405.html?CMP=ILC-001](http://newsroom.cisco.com/dlls/2005/hd_050405.html?CMP=ILC-001)

- [14] LEASE: [www.ieee-infocom.org/2004/Papers/21\\_1.PDF](http://www.ieee-infocom.org/2004/Papers/21_1.PDF)
- [15] Localización y seguimiento de terminales móviles en entornos *Indoor*. J.J. Astrain, J. Villadangos, A. Córdoba. Departamento de Matemática e Informática. Universidad Pública de Navarra.
- [16] Master Thesis Building a Commodity Location-based Service at Botanic Garden of University of Freiburg. Ludmila Salaur. Communication Systems Department. University of Freiburg.
- [17] 802.11- IEEE - <http://www.ieee802.org/11/>
- [18] 802.11 Medium Access Method. Jim Geier.  
<http://www.wi-fiplanet.com/tutorials/article.php/1548381>
- [19] 802.11e –Wikipedia <http://en.wikipedia.org/wiki/802.11e>
- [20] Place Lab: Device Positioning Using Radio Beacons in the Wild - [www.placelab.org](http://www.placelab.org)  
<http://www.placelab.org/publications/pubs/pervasive-placelab-2005-final.pdf>
- [21] Programación para la Detección y Localización de Usuarios Móviles en Redes Inalámbricas Heterogéneas. I. V. Murillo, R. Baltazar. Maestría en Ciencias en Ciencias de la Computación, Instituto Tecnológico de León.
- [22] RADAR: An In-Building RF-based User Location and Tracking System. Bahl, P. et al. Microsoft Corp.  
<http://research.microsoft.com/~padmanab/papers/infocom2000.pdf>
- [23] Redes WiFi <http://technet.microsoft.com/es-es/library/bb457012.aspx>
- [24] Sistema de Guiado Basado en Localización de Dispositivos Móviles en WLAN. Fernando Delicado Huelva. Universidad Carlos III de Madrid.
- [25] Sistema de Localización en interiores. Luis Diaz-Ambrona Tabernilla. Universidad Politécnica de Madrid.
- [26] Sistema de Posicionamiento Global (G.P.S.). Leonardo Casanova M
- [27] Sistema de Posicionamiento Global (GPS): Descripción, Análisis de Errores, Aplicaciones y Futuro. A.Pozo-Ruz, A.Ribeiro, M.C.García-Alegre, L.García, D.Guinea, F.Sandoval. Instituto de Automática Industrial Consejo Superior de Investigaciones Científicas.
- [28] Skyhookwireless <http://www.skyhookwireless.com/>



[29] Tecnologías de Localización. **Ana Bernardos**, U. Carlos III para el Centro de Difusión de Tecnologías ETSIT-UPM

[30] Tema 9 Triangulación y Trilateración. Topografía II. Mercedes Farjas Abadía. Universidad Politécnica de Madrid.

[31] Understanding GPS: Principles and Applications. Kaplan.

[32] Wireless LAN Location System. Johnny Shih. School of Information Technology and Electrical Engineering, The University of Queensland.

## ***Apéndice A: Glosario***

**AC** – Access controller; controlador de acceso. Encaminador IP de una red inalámbrica que suele funcionar, también, como punto de acceso (véase AP).

**ACK** – Acknowledgement; confirmación. Se conoce con este nombre genérico a un mensaje que es enviado desde un receptor a un emisor para confirmar que lo transmitido por este último se ha recibido correctamente.

**ACL** – Access Control List; lista de control de acceso. Mecanismo de autenticación de usuarios basada en una lista blanca de identificadores únicos. Usualmente, esta lista se basa en direcciones MAC para redes inalámbricas en modo infraestructura (véase infraestructura), existiendo más soluciones en modo ad-hoc (véase ad-hoc, OpenSystem).

**Ad-hoc** – Literalmente: “para esto”. En tecnología de redes se aplica este nombre a las redes en las que no hay puntos de acceso centrales (véase Infraestructura).

**ADSL** – Asymmetric Digital Subscriber Line. Tecnología de transmisión de datos a través de un cable de cobre que concede mucho más ancho de banda en el sentido centralita a teléfono (descendente) que en el sentido teléfono a centralita (ascendente). Los anchos de banda alcanzables con ADSL pueden ser de hasta 24 Mbit/s en sentido descendente y 3.5 Mbit/s en sentido ascendente.

**AFLT** – Advanced Forward Link Trilateration; Trilateración avanzada de enlace hacia delante. Sistema de localización basado en comparar las señales de distintas estaciones base en un mismo móvil. (Véase trilateración).

**A-GPS** – Assisted GPS; Sistema asistido por GPS (véase GPS). Extensión del sistema GPS que mejora la localización ofrecida por dicho sistema teniendo en cuenta información disponible en la red móvil a la que pertenece el receptor A-GPS.

**AP** – Access point; punto de acceso a una red inalámbrica.

**AOA** – Angle of Arrival; ángulo de llegada. Sistema para, dado un conjunto de antenas, determinar la dirección o distancia de un objeto basándose en la diferencia entre los tiempos de llegada a cada antena de la misma señal (véase TDOA).

**Autoregresivo** – Modelo matemático de proceso aleatorio que se utiliza en teoría de la señal para simular distintos efectos (como, por ejemplo, para estimar la potencia de una señal transmitida).

**Blackberry** – Literalmente, “zarzamora”. Nombre comercial de una serie de PDAs (véase PDAs) de la compañía canadiense RIM.

**Bluetooth** – Literalmente, “diente azul”, por el apodo del rey Haroldo I de Dinamarca. Sistema para comunicación inalámbrica en distancias muy cortas.

**BSA** – Basic Service Area. Área de servicio básico, es la zona donde se comunican las estaciones de una misma BSS, se definen dependiendo del medio.

**BSC** – Base Station Controller.

**BSD** – Berkley Software Distribution, Distribución de software de Berkley. Nombre genérico de sistemas de licencia gratuita, estilo Unix.

**BSS** – Base Station Subsystem.

**CDMA** – Code division multiple access, Multiplexación por división de código. Sistema de acceso a canal común basado en dos técnicas distintas: codificación de señal mediante código y espectro expandido. Da nombre también al conjunto de protocolos de segunda generación de telefonía móvil que usan esta técnica de multiplexación, promulgados por la telecommunication industry association (TIA) y de uso mayoritario en EEUU y otros países (véase GSM).

**Coordenadas** - Puntos que identifican las zonas dentro del mapa.

**CSMA** – Carrier Sense Multiple Access; Acceso múltiple por detección de portadora. Método para compartir un canal común en el que, antes de transmitir, se comprueba que no hay otro emisor transmitiendo previamente, en cuyo caso se transmite con una probabilidad definida de antemano o aleatoria.

**Cell-ID** – Identificador de celda, disponible en redes móviles de segunda y tercera generación.

**Constelación (de satélites)** – Conjunto de satélites dispuestos en el espacio exterior de forma tal que se cubra un área geográfica dada.

**DCF** – Distributed Coordination Function; Función de coordinación distribuida. Técnica para compartir un canal común, muy similar a CSMA, y considerada una variante de ésta última, en la que, tras una colisión (dos o más emisores intentando transmitir simultáneamente), estos esperaran un tiempo aleatorio antes de volver a intentarlo.

**D-GPS** – Diferencial GPS: Sistema GPS diferencial (véase GPS). Extensión del sistema GPS en el que los receptores reciben información de corrección adicional que permite mejorar la precisión del sistema GPS.

**DSSS** – Direct Sequence Spread Spectrum; Espectro ensanchado por secuencia directa. Técnica de transmisión en un canal radio mediante el cual se repite la señal en todo el ancho de banda, pero combinada con pseudo-ruido, dando lugar a un espectro en potencia parecido al de ruido blanco.

**EAP** – Extended Authentication Protocol; Protocolo extensible de autenticación. Conjunto de técnicas definidas como parte del estándar inalámbrico (véase IEEE) para permitir la autenticación de un usuario. El estándar WPA (véase WPA) se basa en EAP.

**E-OTD** – Enhanced Observed Time Difference – Sistema mejorado de observación de diferencias en el tiempo (véase TDOA). Algoritmo para determinar la posición de un móvil comparando la potencia y el retraso con el que su señal llega a distintas estaciones base.

**ETSI** – European Telecommunication Standards Institute – Instituto europeo de estándares para la telecomunicación. Autor, entre otros, de GSM (véase GSM, GPRS), y de HiperLAN2 (véase HiperLAN).

**FreeBSD** – Variante de BSD (véase BSD), distribuida para la arquitectura x86 de Intel.

**FHSS** – Frequency-hopping spread spectrum; espectro ensachado por salto en frecuencia. Técnica de transmisión en un canal radio en el que la portadora se repite periódicamente en todo el ancho de banda disponible en el canal por el que se transmite.

**FSK** – Frequency-shift keying; Modulación por desplazamiento de frecuencia. Mecanismo de transmisión de señal digital basado en el cambio de frecuencia de la señal portadora.

**Galileo** – Nombre del sistema de posicionamiento global por satélite desarrollado por la Unión Europea (véase GLONASS, GPS).

**GLONASS** – GLObal'naya NAVigatsionnaya Sputnikovaya Sistema; Sistema global de navegación por satélite. Desarrollado y mantenido por Rusia e India (véase GPS, Galileo).

**GNSS** – Sistema Global de Navegación por satélite (Véase Galileo, GLONASS, GPS).

**GPRS** – General packet radio service; Servicio general de paquetes vía radio. Parte del estándar GSM (véase GSM) para ofrecer un servicio de transmisión de datos a una velocidad razonable.

**GPS** – Global Positioning System, sistema de posicionamiento global basado en la señal emitida por un conjunto de satélites. Desarrollado y mantenido por el ejército de EEUU (véase GLONASS, Galileo).

**GSM** – Global system for mobile communications – Sistema global para comunicaciones móviles. Conjunto de protocolos de segunda generación de telefonía móvil promulgados por el European Telecommunications Standard Institute (ETSI), de uso mayoritario en el mundo (véase CDMA, ETSI).

**HiperLAN2** – Protocolo para comunicaciones en un WLAN, desarrollado por la ETSI (véase ETSI).

**Huella Multitrayecto** – Conocida también por su nombre en inglés, Multipath Fingerprint, es la “huella” que se puede obtener comparando los retardos y variaciones de una misma señal al llegar por distintos caminos al mismo punto.

**IEEE** – Institute of Electrical and Electronics Engineers; instituto de ingenieros eléctricos y electrónicos. Asociación de EEUU, especialmente famosa por los estándares que promulga; de entre dichos estándares, en este proyecto cabe destacar la serie 802.x, referidos a WLAN (véase WLAN).

**INEM** – Instituto Nacional de Empleo, antiguo nombre, aún conservado en el uso cotidiano, del Servicio Público de Empleo Estatal (SPEE), organismo encargado de gestionar el empleo en España.

**Infraestructura** – Modalidad de redes inalámbricas en las que hay tanto puntos de acceso como controladores de acceso (véase AC, AP, ad hoc).

**IrDA** – Infrared data association. Organización que define el conjunto de protocolos para comunicación mediante de ondas infrarrojas. Por extensión, también se conoce con este nombre tanto a los protocolos que permiten dicha comunicación como a la comunicación en sí.

**JAVA** – Lenguaje de programación desarrollado por Sun Microsystems.

**J2ME** – Java Platform, Micro Edition. Subconjunto de Java (véase JAVA), especialmente apropiado para dispositivos móviles.

**LAN** – Local Area Network, i.e., red de área local.

**Linux** – Sistema operativo basado en licencia libre, desarrollado por un conjunto de voluntarios liderados por Linus Torvalds.

**Location fingerprinting** – Huella de localización; es un sistema de localización basado en comparar la potencia y el retardo de una señal con una base de datos que guarda dichas potencias y retardos para distintas localizaciones. Se aplica tanto en redes inalámbricas como en redes móviles (véase Huella multitrayecto).

**Localización** - Lugar donde se quiere posicionar a un usuario. Puede ser la planta de una universidad, de un centro comercial, de tu casa.

**MAC** – Media Access Control; Control de acceso al medio (o canal común). Nombre genérico usado para la técnica por la cual se decide como decidir quién tiene derecho a usar un canal común (véase, como ejemplo, CSMA o DCF, PCF). Por extensión, también se llama así a un identificador único de las tarjetas de red, sea red alámbrica o inalámbrica.





**Mapa** - Croquis de la localización donde queremos buscar al usuario.

**NAVSTAR** – Nombre del programa GPS dentro del Ejército de EE.UU. El uso mayoritario, sin embargo, se decanta por GPS en vez de NAVSTAR.

**OFDM** – Orthogonal frequency-division multiplexing; multiplexación por división de frecuencias ortogonales. Técnica para compartir un canal común entre varios emisores, simultáneamente, por el cual las señas de los emisores se dividen en múltiples fragmentos, cada uno de los cuales se transmite usando una portadora ortogonal al resto.

**Open System** – Sistema de autenticación de usuarios en una red *ad-hoc* (véase *ad-hoc*), por el cual sólo se permite la autenticación de aquellos usuarios presentes en una lista blanca.

**Palm OS** – Sistema operativo utilizado por las PDAs (véase PDA) de la compañía Palm.

**PCF** – Point Coordination Function – Función de coordinación a través de un punto. Técnica para permitir el acceso a un canal común en la que todo nodo que desee transmitir por el nodo común ha de esperar a recibir su turno (o su conjunto de frecuencias, dependiendo de si la multiplexación es el tiempo o en la frecuencia) de manos de una estación base. (Véase DCF).

**PDA** – Personal Digital Assistant; Ayudante Personal Digital. Ordenador de bolsillo, con múltiple variedad de modelos y funcionalidades.

**Pocket PC** – Literalmente, ordenador personal de bolsillo. Se conoce así a las PDAs que funcionan con el sistema operativo Windows Mobile (véase PDA, Windows, Windows Mobile).

**Pseudos-distancia** – Distancia obtenida por receptores de un sistema de posicionamiento por satélite antes de corregir el error causado por la falta de sincronía entre los relojes del receptor y los satélites.

**PSK** – Phase-shifting keying; modulación por desplazamiento de fase. Sistema de transmisión en el cual la información se transmite como un cambio de fase de la señal portadora.

**RADIUS** – Remote Authentication Dial in User Service; Servicio de autenticación de usuarios remotos conectados. Protocolo que ofrece capacidades de autorización, autenticación y gestión de usuarios que pretenden conectarse a una red.

**RST/CTS** - Mecanismo para evitar colisiones en el acceso a un medio común, por el cual un nodo que se desea transmitir un mensaje envía una petición de transmisión (Request to Send) y espera a recibir una confirmación de dicha petición (Clear To Send) antes de transmitir. Cualquier nodo que no siendo ni el receptor ni el



emisor del mensaje, reciba un RST o un CTS, esperará antes de intentar realizar una transmisión propia.

**Shared Key** – Contraseña compartida; modo de autenticación en una red ad-hoc (véase ad-hoc), por el cual se autenticara a aquellos usuarios que conozcan una contraseña común.

**SS** – Signal Strength; potencia de señal, aplicado en este proyecto a la potencia de una señal en una red inalámbrica.

**Symbian** – Sistema operativo para dispositivos móviles, desarrollado por la compañía del mismo nombre, actualmente propiedad de Nokia.

**TDOA** – Time difference of Arrival – Diferencia de tiempos de llegada a distintas antenas para la misma señal (véase AOA).

**Triangulación** – Método matemático para localizar posiciones relativas, basado en la geometría de los triángulos, y, específicamente, en las relaciones entre distintos ángulos (véase Trilateración).

**Trilateración** – Método matemático para localizar posiciones relativas, basado en la geometría de los triángulos, y que, al contrario que la triangulación, usa medidas de puntos de referencia (véase Triangulación).

**UMTS** – Universal Mobile Telecommunications System; Sistema universal de telecomunicaciones móviles. Sucesor del estándar GSM para la tercera generación de telefonía móvil (véase GSM).

**U-TDOA** – Uplink TDOA (véase TDOA), TDOA en enlace ascendente. Aplicación de los principios de TDOA a la comunicación en el sentido ascendente en una red móvil, es decir, a la comunicación enviada por el móvil a la estación base.

**VPN** – Virtual Private Network; Red Privada Virtual. Una red a la que se pueden conectar usuarios sólo tras una autenticación. Se utiliza normalmente en el ámbito empresarial para permitir a sus trabajadores acceder a la red interna a través de una conexión a Internet.

**WAF** – Wall Attenuation Factor; factor de atenuación de muros. Modelo matemático para calcular el efecto en una señal de la atenuación causada por paredes.

**WEP** – Wired Equivalent Privacy; privacidad equivalente a línea fija. Protocolo de encriptación de datos transmitidos en una red inalámbrica, basado en el algoritmo RC4. Tiene vulnerabilidades conocidas y se considera poco seguro. (Compárese con WPA).

**WiFi** – Término comercial de la Wireless Alliance, usado para denotar una red inalámbrica, y, que, según esta alianza, no significa nada, siendo sólo una marca



registrada. Algunos autores sugieren que el nombre pretende hacer pensar en Wireless-Fidelity, por analogía con High-Fidelity (Hi-fi).

**Windows** – Sistema operativo desarrollador por Microsoft Corporation.

**Windows Mobile** – Versión de Windows (véase Windows) especialmente diseñada para dispositivos portátiles.

**WLAN** – Wireless LAN; Red de área local inalámbrica (véase LAN, IEEE, HiperLAN).

**WPA** – WiFi Protection Access; acceso protegido WiFi (véase WiFi). Protocolo de encriptación de datos transmitidos en una red inalámbrica, basado en el algoritmo AES. (Compárese con WEP, véase EAP).

**WPS** – WiFi Positioning System; sistema de posicionamiento WiFi. Un sistema desarrollado por la compañía Skyhook, en el que se intenta combinar la localización por GPS y por redes WiFi (véase GPS, WiFi).

**Zona** - Regiones en las que se subdivide la localización, y donde podemos posicionar al usuario móvil.

## *Apéndice B: Manual de Despliegue del Servidor*

Para poder desplegar el sistema se deben seguir los siguientes pasos:

- Instalación en la máquina que hará de servidor:
  - Jre 1.6.0.07 o superior.  
Desde <http://developers.sun.com/downloads>
  - Conector JDBC en Java\jre1.6.0\_07\lib\ext\  
**mysql-connector-java-5.1.6-bin.jar**  
Desde: <http://dev.mysql.com/downloads/connector/j/5.1.html>
  - MySQLServer 5.0  
Desde: <http://dev.mysql.com/downloads/mysql/5.0.html>
- Despliegue del sistema:
  - Crear la base de datos a partir del esqueleto proporcionado wifitool.sql  
  
Ejecutar el siguiente comando desde línea de comandos en el directorio:  
  
c:\Archivos de Programa\MySQL\MySQLServer5.0\  
  
mysql wifitool < wifitool.sql
  - Copiar el directorio dist que se encuentra en el directorio D:\System\WifiTool del CD a la carpeta destino del sistema, por ejemplo, c:\Wifitool\dist\
- Ejecución:  
  
Desde la carpeta destino del sistema:
  - Realiza doble click en el fichero WifiTool.jar
  - o desde línea de comandos java -jar WifiTool.jar

## ***Apéndice C: Manual de Usuario del Cliente Gráfico de Prueba.***

- Despliegue del sistema:
  - Copia el directorio dist que se encuentra en D:\System\WiLBSCClient del CD a la carpeta destino del sistema, por ejemplo, c:\WiLBSCClient\dist\

- Ejecución:

Desde la carpeta destino del sistema:

- Realiza doble click en el fichero WiLBSCClient.jar
- o desde línea de comandos java -jar WiLBSCClient.jar

- Funcionamiento:

Se pueden elegir tres tipos de funcionamiento:

- Gauge

- LIST

En esta pestaña se pueden visualizar las localizaciones disponibles en el Sistema WifiTool.

- CONFIGURATION

Esta pestaña sirve para configurar la localización por primera vez. Se debe:

- Elegir un nombre, rellenar el campo “Location\_name”
- Añadir un mapa (.gif o .jpeg) que identifique físicamente la localización. Con el botón Add Map File.
- Añadir las coordenadas del mapa. Coordenadas que definan cuadrados dentro del mapa: xa,ya;xb,yb

Una vez rellenos todos los campos se debe enviar la información al sistema mediante el botón “OK”.

- CALIBRATION

En esta pestaña se pueden calibrar las localizaciones existentes en el sistema zona a zona.

Para ello se debe en primer lugar seleccionar la zona a calibrar mediante las opciones que aparecen en el desplegable de Localización, y una vez seleccionada la localización seleccionar una de las zonas que aparecen disponible en dicha localización y pulsar el botón de “OK”.

- DELETE

En esta pestaña se pueden borrar las localizaciones y sus calibraciones.

Para ello se selecciona la localización a borrar y se pulsa el botón “DELETE”.

- User/Mobile

- LOCATION

Permite localizar al usuario en una localización. Para ello se selecciona la localización en la que se encuentra y se pulsa el botón “LOCATE”. Entonces el sistema mostrará su posición.

- Monitor

- LOCATION

Permite localizar los usuarios registrados en una localización. Para ello se selecciona la localización de la que se quiere averiguar la posición de sus usuarios y se pulsa el botón “LOCATE”. Entonces el sistema mostrará su posición.



## ***Apéndice D: Manual de Usuario del Cliente Texto de Prueba.***

- Despliegue del sistema:
  - Descargar el ejecutable del cliente:
    - dist\WifiToolClient.jar
- Ejecución:
  - java -jar dist\WifiToolClient.jar
- Funcionamiento:
  - CONNECTION
  - INITCAL
  - CAL
  - LIST
  - LOC
  - SETLOC
  - GETALLLOC
  - ZONE

## ***Apéndice E: Manual de Creación de Cliente para el Sistema.***

Manual para poder implementar clientes que se comuniquen con el Sistema de Localización WifiTool.

### **Comandos soportados en el servidor**

El servidor no es sensible a mayúsculas, todos los comandos serán interpretados en mayúsculas o minúsculas indistintamente, tanto como si tiene todas las letras en mayúsculas o solo tiene algunas, ídem para las minúsculas.

Los comandos soportados tendrán el formato y número de parámetros mostrado a continuación.

### **Comunes para todos los clientes**

- LIST

Formato: LIST

Petición de las localizaciones disponibles en el servidor.

- BYE

Formato: BYE

Petición de desconexión con el servidor.

- ZONE

Formato: ZONE <loc>

<loc> String con el nombre de la localización a calibrar.

Petición del número de zonas en las que se encuentra dividida la localización correspondiente.





## Cliente Gauge

Solo para el cliente “Gauge” o calibrador, son los comandos que se usan para calibrar el sistema:

- INITCAL

Formato:

INITCAL <loc> <map> <coor>

<loc> String con el nombre de la localización a calibrar.

<map> String con el nombre del mapa (.gif o similar) de la Localización.

<coor> Coordenadas en las que se dividen las zonas del mapa.

Formato:

xa1;ya1;xb1;yb1;.....;xaN;yaN;xbN;ybN

- CAL

Formato:

CAL <loc>> <MAC1> <Pot1> ... <MACN> <PotN>

<loc> String con el nombre de la localización a calibrar.

<MAC1> <Pot1> ... <MACN> <PotN>

Secuencia de parejas de direcciones MAC detectadas junto con la Potencia en db que se recibe de ellas.

## Cliente USER/MOBILE

Solo para el cliente “User” o móvil, son los comandos que se usan para saber la posición del usuario:

- LOC

Formato:

LOC <loc> <userMac> <MAC1> <Pot1> ... <MACN> <PotN>

<loc> String con el nombre de la localización a calibrar.

<MAC1> <Pot1> ... <MACN> <PotN>

Secuencia de parejas de direcciones MAC detectadas junto con la Potencia en db que se recibe de ellas.

- OK\_REQ

Comunica al Servidor que ha recibido bien los datos de la localización y que espera recibir el fichero que contiene el mapa de la misma.

- OK\_ZONE

Solicita saber los datos de la zona en la que se encuentra, pues ya ha recibido todos los datos necesarios para identificarla.

- SETLOC

Formato:

SETLOC <userMac> <MAC1> <Pot1> ... <MACN> <PotN>

<userMac > Dirección MAC del usuario.

<MAC1> <Pot1> ... <MACN> <PotN>

Secuencia de parejas de direcciones MAC detectadas junto con la Potencia en db que se recibe de ellas.

## Cliente Monitor

- GETALLLOC

Formato: GETALLLOC <loc>

<loc> String con el nombre de la localización a calibrar.

- OK\_REQ

Comunica al Servidor que ha recibido bien los datos de la localización y que espera recibir el fichero que contiene el mapa de la misma.

- OK\_ZONE

Solicita saber los datos de la zona en la que se encuentran los usuarios.

## Respuestas del servidor

- ERROR: “Error message”

La aplicación devuelve la palabra error, seguida de un mensaje indicando el error ocurrido siempre que se produzca un error.

Mensajes de error que puede enviar:

- OK:

La aplicación comunica que ha completado una acción correctamente. Existen diversos casos:

- Client-Server connection

- USER

Pregunta por el nombre del usuario (en este momento user). Espera que el cliente le devuelva el usuario.

- PASSWD

Pregunta por la contraseña del usuario (en este momento passwd). Espera que el cliente le devuelva la contraseña.

- REQ-MAP <loc>

La Aplicación solicita que comience el envío fichero que contiene el mapa de la localización <loc>.

- ACK:

Manda un ACK para decir que todo se ha realizado correctamente. Existen diversos casos:

- New loc created.

Manda un ACK para decir que todo se ha realizado correctamente.

- SEND

Formato: SEND <nMap> <coor> <nZone>



- Zone

Formato: Zone <zone>

Devuelve el número de zona en al que se encuentra el usuario.

- Locations

Formato: Locations: loc1; loc2; .... ; locn

Envía las localizaciones disponibles en el sistema separadas por un punto y coma.

## Secuencias de comandos

- LIST

Formato:

LIST

Return:

Locations: loc1; loc2; .... ; locn

- BYE

Formato:

BYE

Return:

Cierre del socket.

- Establecimiento de sesión

Server:

OK: Client-Server connection

USER

Pregunta por el nombre del usuario. Espera que el cliente le devuelva el usuario.

PASSWD

Pregunta por la contraseña del usuario (en este momento passwd). Espera que el cliente le devuelva la contraseña.

OK

- ERROR: “Error message”

La aplicación devuelve la palabra error, seguida de un mensaje indicando el error ocurrido siempre que se produzca un error.

- INITCAL

Formato:

INITCAL <loc> <map> <coor>

<loc> String con el nombre de la localización a calibrar.  
<map> String con el nombre del mapa (.gif o similar) de la Localización.  
<coor> Coordenadas en las que se dividen las zonas del mapa.  
Formato:

Return:

REQ-MAP <loc>

La Aplicación solicita que comience el envío fichero que contiene el mapa de la localización <loc>.

Cliente envía el fichero a través de los sockets

ACK: New loc created.

Manda un ACK para decir que todo se ha realizado correctamente.

- CAL

Formato:

CAL <loc>> <MAC1> <Pot1> ... <MACN> <PotN>

<loc> String con el nombre de la localización a calibrar.  
<MAC1> <Pot1> ... <MACN> <PotN>

Return:

ACK: New data inserted.

- LOC

Formato:

LOC <loc> <userMac> <MAC1> <Pot1> ... <MACN> <PotN>

<loc> String con el nombre de la localización a calibrar.



<MAC1> <Pot1> ... <MACN> <PotN>

Return:

SEND <nMap> <coor> <nZone>

Client:

OK\_REQ

SERVER: envía el fichero que contiene el mapa

Client: OK\_ZONE

Solicita la zona

SERVER:

Zone: <zone>